RADC-TR-89-259, Vol VII (of twelve)
Interim Report
October 1989

NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT - 1988 Research in Automated Photointerpretation

Syracuse University

Dr. J.W. Modestino

APPROVED FOR PUBLIC RELESE; DISTRIBUTION UNLIMITED.

This effort was funded partially by the Laboratory Director's fund.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS) At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-89-259, Vol VII (of twelve) has been reviewed and is approved for publication.

APPROVED: Lee A. Ruiu

LEE A. RUIU
Project Engineer

APPROVED: Walter Senus

WALTER J. SENUS
Technical Director
Directorate of Intelligence & Reconnaissance

FOR THE COMMANDER: Igor G. Plonisch

IGOR G. PLONISCH
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS
N/A

2a. SECURITY CLASSIFICATION AUTHORITY
N/A

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
N/A

5. MONITORING ORGANIZATION REPORT NUMBER(S)
RADC-TR-89-259, Vol VII (of twelve)

6a. NAME OF PERFORMING ORGANIZATION
Northeast Artificial
Intelligence Consortium (NAIC)

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
Rome Air Development Center (COES)

6c. ADDRESS (City, State, and ZIP Code)
Science & Technology Center, Rm 2-296
111 College Place, Syracuse University
Syracuse NY 13244-4100

7b. ADDRESS (City, State, and ZIP Code)
Griffiss AFB NY 13441-5700

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Rome Air Development Center

8b. OFFICE SYMBOL (If applicable)
COES

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
F30602-85-C-0008

8c. ADDRESS (City, State, and ZIP Code)
Griffiss AFB NY 13441-5700

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO
62702F	5581	27	13

11. TITLE (Include Security Classification)
NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT - 1988
Research in Automated Photointerpretation

12. PERSONAL AUTHOR(S)
Dr. J. W. Modestino

13a. TYPE OF REPORT
Interim

13b. TIME COVERED
FROM Jan 88 TO Dec 88

14. DATE OF REPORT (Year, Month, Day)
October 1989

15. PAGE COUNT
132

16. SUPPLEMENTARY NOTATION This effort was funded partially by the Laboratory Directors' Fund. This effort was performed as a subcontract by Rensselaer Polytechnic Institute to Syracuse University, Office of Sponsored Programs.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Artificial Intelligence
Expert Systems
Machine Vision
Photointerpretation
Image Analysis

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose is to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress that has been made in the fourth year of the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system.

The specific topics for this volume are the use of expert systems for automated photo interpretation and other AI techniques to image segmentation and region identification.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Lee A. Ruiu

22b. TELEPHONE (Include Area Code)
(315) 330-4863

22c. OFFICE SYMBOL
RADC (IRRE)

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Item 10. SOURCE OF FUNDING NUMBERS (Continued)

Program Element Number	Project Number	Task Number	Work Unit Number
62702F	5581	27	23
61102F	2304	J5	01
61102F	2304	J5	I5
33126F	2155	02	10
61101F	LDFP	27	01

NAIC

Northeast Artificial Intelligence Consortium

ANNUAL REPORT 1988

VOLUME 7

RENSSELAER POLYTECHNIC INSTITUTE

Research in Automated Photointerpretation

Principal Investigator: Dr. J.W. Modestino

This work was supported by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York, 13441-5700, and the Air Force Office of Scientific Research, Bolling Air Force Base, DC, 20332, under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

7.1 Introduction:

The RPI task has been concerned with the development of expert systems techniques for automated photointerpretation. More specifically, our efforts have been directed toward the development, implementation and demonstration of techniques which will mimic the job of a trained photoanalyst in interpreting objects in monochrome, single-frame aerial images. This is a difficult task which requires a combination of numerical and symbolic image processing techniques.

During the course of this effort we have developed a novel hierarchial, region-based approach to automated photointerpretation (cf. [1]). Basically, this approach proceeds by first segmenting the input image into disjoint regions which differ in tonal or textural properties. The spatial relationships between different regions are then expressed in terms of the associated adjacency graph where nodes represent regions and the connectivity indicates regions which are spatially contiguous. Based upon knowledge of the underlying spatial adjacency graph, together with various self and mutual region attributes or features, the problem is then that of assigning interpretations, or object categroies, to each of the nodes. This is generally a computationally explosive task. The novelty of our approach is that we have been able to develop a computationally feasible approach to this symbolic interpretation process.

The advantage of our approach is based upon two important properties: First, we model the interpretation process as a Markov random field (MRF) defined on the adjacency graph. Secondly, we make use of an efficient stochastic relaxation process to find the most likely interpretation. The first assumption allows us to localize the search for good interpretations while the second helps in avoiding the otherwise computationally explosive nature of the search for optimum interpretations.

Our major effort during FY'88 has been in refining the region hierarchial approach, improving the initial segmentation process and, finally, demonstrating the approach on real-world aerial photographs. The present report is an attempt to document this progress of the last year.

This final report is organized as follows: In Section 7.2 we provide a detailed description of the current states of our hierarchial, region-based approach to automated photointerpretation. This is followed, in Section 7.2 by a detailed development of an unsupervised model-fitting approach to cluster validation with particular application to image segmentation. This has been used in our image interpretation approach. In Section 7.3 we describe an implementation of this overall image interpretation approach on the TI Explorer.

References for Section 7.1

1. J.W. Modestino, "A Hierarchial Region-Based Approach to Automated Photointerpretation," NAIC Final Report for FY'86.
2. H.L. Van Trees, *Detection, Estimation and Modulation Theory I* Wiley and Sons, New York, 1968.
3. R. Kinderman and J.L. Snell, *Markov Random Fields and Their Applications,* American Mathematical Society, Providence, RI, 1980.

7.2 A Markov-Random Field Model-Based Approach to Image Interpretation:

7.2.1 Introduction:

In this section, a Markov random field (MRF) model-based approach to automated image interpretation is described and demonstrated. This scheme is a region-based approach in which an image is first segmented into a collection of disjoint regions which form the nodes of an adjacency graph. Once the adjacency graph has been determined, image interpretation is achieved through assigning object labels, or interpretations, to the segmented regions, or nodes, using domain knowledge, extracted feature measurements and spatial relationships between the various regions. In this approach, the interpretation labels are modeled as a MRF on the corresponding adjacency graph and the image interpretation problem is then formulated as a maximum a posteriori (MAP) estimation rule given domain knowledge and region-based measurements. Simulated annealing is used to find this best realization, or optimal MAP interpretation. Through the MRF model, and its associated Gibbs distribution, this approach also provides a systematic method for organizing and representing domain knowledge through appropriate design of the clique functions describing the Gibbs distribution representing the pdf of the underlying MRF. We provide a general methodology for design of the clique functions. Results of image interpretation experiments performed on synthetic and real-world images using this approach are described and appear promising.

7.2.2 Background:

Image interpretation is the process of understanding the meaning of an image through identifying significant objects in the image and analyzing their spatial relationships. These objects can be very simple, such as tools and work parts in an assembly line scene; or they can be quite complicated and composed of many simpler objects, such as a runway area full of airplanes in an airport scene.

The need for image interpretation can be found in many diverse fields of science and engineering. For example, a major application of image interpretation is in remote-sensing, or aerial/satellite photointerpretation, which is widely used in geological survey and military air reconnaissance [1]- [3]. Image interpretation also plays an important part in biomedical science and particle physics where much of the experimental results are recorded in the form of photographs [4]-[5].

Traditionally, the task of image interpretation is performed by well-trained and experienced *human experts*. However, analyzing a complex image is quite labor intensive. Furthermore, as a result of the rapid advances in imaging/photographic technology, in many of the previously mentioned applications the large amount of images generated would soon overload the relatively small number of experts. Hence, much of the research in image processing has been directed towards constructing *automated* (computerized) image interpretation systems. Recent research in intelligent robots has created yet another need for automated image interpretation. In this case, the robots need to understand what they "see" with imaging sensors in order to be able to perform intelligent tasks in complex environments [6]-[7]. Here, the robots have to rely entirely on automated image interpretation.

Most of the existing image interpretation techniques involve two major operations, *low-level* and *high-level* processing. In low-level processing, the representation of an image is transformed, through image processing operations, such as edge detection and region segmentation, from a *numerical* representation, as an array of pixel intensities, to a *symbolic* representation, as a set of spatially related *image primitives*, such as edges and regions. Various features are then extracted from the primitives. These features may include: the lengths of significant edges, average intensities of regions, shape and/or texture descriptors, etc. Also extracted would be the spatial relationship between the image primitives. In high-level processing, image domain knowledge is used to assign object labels, or interpretations, to the primitives and construct a description as to "what is present in the image". In the rest of this paper, we often refer to the object labels as interpretations and the overall interpretations for *all* the primitives in the image as the interpretation of the image.

The main approach in early research on image interpretation was that of classification [4]-[5], [8] in which isolated image primitives are classified into a finite number of object classes according to their feature measurements. However, since low-level processing often produces erroneous or incomplete primitives and noise in the image may often cause measurement errors in the features, the performance of image interpretation systems using the classification approach is quite limited. Tne main problem here is that the rich knowledge of the spatial constraints between objects, used by human experts, has not been used in the high-level processing.

To solve this problem, most of the recent techniques have adopted the *knowledge-based*, or *expert system*, approach. In this approach, domain knowledge, and especially spatial constraints, are used in high-level (some also in low-level) processing. Hence, an ambiguous object may be recognized as the result of successful recognition of its neighboring objects. Even more fundamentally, an object can be recognized from combining the feature information from several spatially-related image primitives. Finally, low- level processing errors may be corrected, or at least mitigated, through feedback from high-level processing to low-level image processing.

The early work in knowledge-based image interpretation has been summarized in Nagao and Matsuyama [11], Binford [12] and Ohta [13]. Recently, a number of more sophisticated experimental systems have been constructed for different application domains, such as high-altitude aerial photographs [11], [14]-[15], [16]-[18]; airport scenes [19], [20]-[21] and outdoor scenes [13],[22]-[24]. Many of these systems are still undergoing continuous improvements through architecture modification and domain extension. New ideas and systems are constantly emerging, as can be seen in recent PRCV and SPIE conferences and workshops, and several pertinent technical reports [25]-[26],[21]. While success has been demonstrated to various degrees in these systems, developing a *general*, *domain-independent* and *systematic* method for constructing knowledge-based image interpretation systems is still an open problem [22].

In this paper, we describe a general, domain-independent, stochastic model- based approach to the image interpretation problem. In this approach, the interpretation labels to be assigned to the primitives of an image are modeled as a Markov random field (MRF) defined on the spatial adjacency graph formed by the primitives, where the randomness is used to model the uncertainty in the assignment of the labels. As a result, the domain knowledge, whatever it may be, can be systematically represented in terms of the clique functions associated with the underlying Gibbs probability distribution function (pdf) describing the MRF. Under the MRF modeling assumption, image interpretation is then formulated as the optimization problem of maximizing the a posteriori probability of interpretation given domain knowledge and feature measurements. Then, simulated annealing is used to find the optimal set of interpretation labels. In this paper, we present a special region-based version of this approach. That is, the primitives are segmented regions; and for the sake of simplicity, we do not include feedback from high-level to low-level processing. However, research is currently ongoing to include use of linear edge segments as primitives as well as high-to-low level feedback [27]-[28]. This will also be discussed in subsequent sections.

This paper is organized as follows. In the next section, we describe the MRF model-based formulation for the image interpretation problem. Then, in Section 7.2.4, we will show how domain knowledge can be organized into clique functions associated with the MRF model. After the discussion of the implementation of the optimization (interpretation) through the simulated annealing procedure in Section 7.2.5, we will present and discuss results of image interpretation experiments performed on synthetic and real-world images in Sections 7.2.6 and 7.2.7. Finally, a summary and directions for future research are provided in Section 7.2.8.

7.2.3 The MRF Model-Based Approach to Image Interpretation:

The MRF model, as an extension of the one-dimensional Markov process, has recently attracted much attention in the image processing and computer vision community. The main advantage of the MRF model is that it provides a general and natural model for the interaction between spatially related random variables and there is a relatively efficient optimization algorithm, simulated annealing, that can be used to find the *globally* optimal realization which, in this case, corresponds to the maximum a posteriori (MAP) interpretation. Up to now, the success of MRF models has been demonstrated mostly in low-level image processing applications, such as region segmentation [29]-[36] and edge detection [37], where they are defined on two- dimensional (2-D) lattices on which the images are represented as 2-D arrays. For example, in stochastic model-based image segmentation, the pixels are classified into a finite number of statistical *classes* and the MRF is used to model the spatial distribution of pixel classes, or region distributions [29]-[36]. However, as demonstrated by Kinderman and Snell [38], the MRF can be defined, in general, on graphs for which the 2-D lattice is a special case. In what follows, we will briefly review the concepts associated with the MRF defined on graphs and show how this can be applied to the image interpretation problem. More comprehensive treatments on MRF's can be found in [38]-[39].

A.) The MRF Model on Graphs:

Let $\mathbf{G} = \{\mathbf{R}, \mathbf{E}\}$ be a graph, where

$$\mathbf{R} = \{R_1, R_2, \ldots, R_N\}, \tag{1}$$

is the set nodes represented by R_i, $i = 1, 2, \ldots, N$; $\mathbf{E}$ is the set of edges connecting them. Suppose that there exists a *neighborhood system* on $\mathbf{G}$, denoted by

$$\mathbf{n} = \{n(R_1), n(R_2), \ldots, n(R_N)\}, \tag{2}$$

where $n(R_i)$, $i = 1, 2, \ldots, N$, is the set of all the nodes in $\mathbf{R}$ that are neighbors of R_i, such that

i.) $R_i \notin n(R_i)$, and

ii.) if $R_j \in n(R_i)$ then $R_i \in n(R_j)$.

Let

$$\mathbf{I} = \{I_1, I_2, \ldots, I_N\} \tag{3}$$

be a family of random variables defined on $\mathbf{R}$. Then, $\mathbf{I}$ is called a *random field*, where I_i is the random variable associated with R_i. Notice that the random variables I_i's here can be numerical as well as symbolic, e.g., interpretation labels. We say $\mathbf{I}$ is a MRF on $\mathbf{G}$ with respect to the neighborhood system $\mathbf{n}$ if and only if

i.) $P[\mathbf{I}] > 0$, for all realizations of $\mathbf{I}$;

ii.) $Pr[I_i | I_j, \text{all } R_j \neq R_i] = P[I_i | I_j, R_j \in n(R_i)]$,

where $P[\cdot]$ and $P[\cdot|\cdot]$ are the joint and conditional pdf's, respectively. Intuitively, the MRF is a random field with the property that the statistics at a particular node depends mainly on that of its neighbors.

An important feature of the MRF model defined above is that its joint pdf has a general functional form, known as the *Gibbs distribution*, which is defined based on the concept of *cliques* [38]-[39]. Here, a clique associated with the graph $\mathbf{G}$, denoted by c, is a subset of **R** such that it contains either a single node or several nodes that are *all* neighbors of each other. If we denote the collection of all the cliques of **G** with respect to the neighborhood system **n** as $\mathbf{C}(\mathbf{G}, \mathbf{n})$, the general functional form of the pdf[1] of the MRF can be expressed as the following Gibbs distribution:

$$P[\mathbf{I}] = Z^{-1} exp[-U(\mathbf{I})], \tag{4a}$$

where

$$U(\mathbf{I}) = \sum_{c \in \mathbf{C}(\mathbf{G},\mathbf{n})} V_c(\mathbf{I}), \tag{4b}$$

is called the *Gibb's energy function* and $V_c(\mathbf{I})$'s are called clique functions defined on the corresponding cliques $c \in \mathbf{C}(\mathbf{G}, \mathbf{n})$. Finally,

$$Z = \sum_{\text{all } \mathbf{I}'} exp\Big[-U(\mathbf{I}')\Big], \tag{4c}$$

is the normalization factor to make (4a) a valid pdf. Notice that the MRF pdf above is quite rich in that the clique functions can be arbitrary as long as they depend only on the nodes in the corresponding cliques. Due to this unique structure, in which the global and local properties are related through cliques, the MRF model-based approach to image interpretation provides potential advantages in knowledge representation, learning and optimization, as will be discussed in more detail later. More importantly, this method provides a useful mathematical framework for the study of image interpretation procedures.

B.) The MRF Model-based Formulation:

As described in Section 7.2.2, for the time being we restrict the image interpretation problem to that of labeling *segmented regions*. Suppose for a given image, there are N disjoint regions after segmentation,[2] denoted by $\mathbf{R} = \{R_1, R_2, \ldots, R_N\}$. Then **R** can be represented by a set of nodes in a connected graph, called the *adjacency graph*, denoted

[1] Actually, this is a probability mass function (pmf) due to the discrete nature of **I** although we will not make this distinction in what follows and continue to use the term pdf.

[2] Clearly, the number N of segmented regions is a random variable depending upon the image as well as the segmentation procedure.

by $\mathbf{G} = \{\mathbf{R}, \mathbf{E}\}$; where the edge set $\mathbf{E}$ is such that a node R_i is connected to another node R_j if and only if the corresponding regions are spatially adjacent. A neighborhood system, denoted by $\mathbf{n}$, can also be defined on the adjacency graph. For simplicity, in what follows we define the neighbors of a node to be the nodes that are connected to it directly by an edge of $\mathbf{G}$, i.e., only spatial adjacent regions are neighbors. Now, given the neighborhood system, we can also find the cliques for the adjacency graph. As an illustration, we have shown in Fig. 7.2.1 the adjacency graph and all its cliques for a particular synthetic conceptual image. This image is intended to represent a car on a road between two fields with the sky as a background. In forming the adjacency graph, we assume perfect segmentation of the image objects.

As described in Section 7.2.2, image interpretation is the process of assigning object labels to the segmented regions according to domain knowledge and feature measurement information (or *measurements*, in short) made on these regions. From the above graphical formulation, the interpretation of the image can be represented as a vector $\mathbf{I}(\mathbf{R}) = \{I_1, I_2, \ldots, I_N\}$, defined on the adjacency graph $\mathbf{G}$, where we use $\mathbf{I}(\mathbf{R})$ to emphasize the relationship between interpretation and the symbolic representation in terms of segmented regions. Here, I_i, $i = 1, 2, ..., N$, is the interpretation label for node R_i; while $I_i \in L$ and $L = \{L_1, L_2, \ldots, L_M\}$ is the set of all the interpretation labels. In addition, we consider I_i's as symbolic random variables to account for the uncertainty in assigning object labels to segmented regions due to, e.g., image noise and segmentation errors. Hence, $\mathbf{I}(\mathbf{R})$ is a random field. Let's denote the domain knowledge as $\mathbf{K}$ and all the measurements made on the segmented regions as $\mathbf{X}(\mathbf{R})$. Now, we can define image interpretation as the following *optimization problem*: for a given $\mathbf{R}$, find $\mathbf{I}_0(\mathbf{R})$, such that

$$\mathbf{I}_0(\mathbf{R}) = \arg \max_{\mathbf{I} \in \{L\}^N} P[\mathbf{I}(\mathbf{R}) \,|\, \mathbf{K}, \mathbf{X}(\mathbf{R})], \tag{5}$$

where $P[\cdot|\cdot,\cdot]$ is the a posteriori pdf of the interpretation given the domain knowledge and measurements, while $\{L\}^N$ is the set of all possible interpretation vectors of length N. The formulation of (5) is also known as the maximum a posteriori (MAP) formulation.

Two problems must be solved in applying the above MAP approach to image interpretation. Specifically, we need an explicit expression for the conditional pdf in (5) and an optimization method to avoid the computationally explosive nature of exhaustive combinatorial search. Feldman and Yakimovski [40], and Faugeras and Price [14]-[15] have considered similar formulations to that of (5) and proposed heuristic expressions for the a posteriori pdf using the marginal pdf's of single and joint pdf's of pairs of interpretation labels. They have also used different relaxation schemes to find local optimal solutions, some of which have also been studied in [54]-[57]. On the other hand, the MRF model discussed in A.) appears to provide a natural solution to the above two problems. More specifically, assume that **I**(**R**) forms a MRF. Then, the pdf appearing in (5) is the Gibbs distribution

$$P[\mathbf{I}(\mathbf{R})\,|\,\mathbf{K},\mathbf{X}(\mathbf{R})] = Z^{-1}exp[-U(\mathbf{I}(\mathbf{R})\,;\,\mathbf{K},\mathbf{X}(\mathbf{R}))], \tag{6a}$$

with energy function

$$U(\mathbf{I}(\mathbf{R})\,;\,\mathbf{K},\mathbf{X}(\mathbf{R})) = \sum_{c\in \mathbf{C}(\mathbf{G},\mathbf{n})} V_c(\mathbf{I}(\mathbf{R})\,;\,\mathbf{K},\mathbf{X}(\mathbf{R})), \tag{6b}$$

where the $V_c(\cdot;\cdot,\cdot)$'s are the clique functions. Indeed, as will be seen in the subsequent sections, through imposing a neighborhood system and the Markov property of ii.), the MRF model-based formulation provides a general and systematic approach for knowledge representation and knowledge acquisition through appropriate construction of the clique functions. For the optimization strategy, the simulated annealing procedure can be used to find the globally optimal interpretation for the image. In addition, the approach of [40],[14],[15] can be shown to be special cases with certain neighborhood structures and clique functions. Finally, when used in the context of image interpretation, the MRF model suggest that the interpretation for a particular region given those of all other regions, depends only on the interpretations of its neighboring regions. This is often a reasonable assumption in practical applications. For example, the identification of a region as a car might depend on whether its neighboring regions are a road but has little to do with the identity of the regions spatially far removed from it. In the rest of the paper, we will model the interpretation vector **I**(**R**) as a MRF. Since simulated annealing is a relatively well-defined procedure, we will concentrate on the knowledge engineering aspects of the image interpretation problem; that is, knowledge representation and learning through constructing the pdf of the MRF.

Finally, we should point out that, although the MRF model-based approach is presented here in the form of a region-based approach, it can be extended to include other primitives and to model the situation where interaction between high-level and low-level processing (e.g., feedback) is used. When other primitives, such as linear edge segments, are introduced they can be considered as nodes of a generalized adjacency graph; they can also be considered as features associated with different regions rather than primitives themselves. To model the high-level and low-level interaction during interpretation, the adjacency graph can be considered as a *dynamic* graph which changes with time; subsequently, the MRF become also a dynamic model. Currently, these problems are under active investigation.

7.2.4 The Design of Clique Functions:

In the MRF model-based formulation of the preceding section, it is clear that the optimal interpretation, $\mathbf{I}_0(\mathbf{R})$, should be the one that minimizes the energy function, or has the minimum energy. For a given image, the optimal interpretation depends on how the energy function is defined. In general, we would like the optimal interpretation obtained under the MRF assumption to be the one that is most *consistent* with the measurements and domain-knowledge. For example, in aerial photointerpretation, suppose we know that a car has small area and would usually be on a road. An interpretation with a car having large area or in the sky should obviously be considered *not* optimal. This type of consistency requirement can be achieved by properly selecting the energy functional or, rather, the corresponding clique functions. It will be seen in the following that, by using the MRF model, the domain knowledge can be organized easily and systematically as clique functions to provide a proper energy functional such that the consistency between the interpretation, the measurements and domain knowledge is maintained.

Without loss of generality, we assume that all the clique functions are non- negative. Then, a general principle for the selection of a clique function is the following.

Design Rule:

If the interpretation of the regions (or region for a *singleton* clique) in a clique tends to be consistent with the measurements and domain knowledge, the clique function decreases, resulting in a decrease in the energy function; otherwise, the clique function increases, resulting in a corresponding increase in the energy function.

In this way, an interpretation for the image that is most consistent with the measurements and domain knowledge will have the minimum energy, or achieve the optimum. Based on this principle, we now propose a general approach to defining clique functions from domain knowledge. We first consider the clique functions for single-node cliques, and then extend the result to the case of multiple-cliques.

A.) Clique Functions for Single-Node Cliques:

Let c be an arbitrary single-node clique with one node, R. Let the corresponding clique function be denoted by $V_c(I(R)\,;\mathbf{K},\mathbf{X}(R))$, it depends only upon the single node R, its interpretation $I = I(R)$, and the measurements $\mathbf{X}(R)$ on the corresponding segmented region R, as well as the domain knowledge represented by $\mathbf{K}$. Suppose that $\mathbf{X}(R)$ has m components, $X_1(R), X_2(R), \ldots, X_m(R)$, representing measurement values of m well-defined *features* of R, e.g., *average gray level, area, standard deviation of gray-levels*, etc. Assuming the components of $\mathbf{X}(R)$ are independent, we can define a clique function for clique c as

$$V_c(I(R)\,;\mathbf{K},\mathbf{X}(R)) = \sum_{i=1}^{m} p_c^{(i)}(I(R),\mathbf{K}) B_c^{(i)}(I(R);\mathbf{K}, X_i(R)), \tag{7}$$

where $B_c^{(i)}(\cdot;\cdot,\cdot)$, $i = 1, 2, \ldots, m$, are called *basis functions* for the corresponding clique function. These quantities are functions of the i'th feature measurement, $X_i(R)$, parameterized by the interpretation $I(R)$ and, of course, depend upon the domain knowledge, $\mathbf{K}$. The $p_c^{(i)}(I,\mathbf{K})$'s, are a set of non negative numbers

$$p_c^{(i)}(I,\mathbf{K}) \geq 0\ ;\ i = 1, 2, \ldots, m, \tag{8a}$$

which can be conveniently normalized so that

$$\sum_{i=1}^{m} p_c^{(i)}(I,\mathbf{K}) = 1, \tag{8b}$$

and are *weights* associated with the basis functions. Here, $p_c^{(i)}(I, \mathbf{K})$ not only depends on i but also on the interpretation $I(R)$ as well as $\mathbf{K}$.

Now the problem of designing clique functions becomes that of designing the basis functions of the features and determining their weights. We first consider the design of the basis functions. Without loss of generality, we assume that all the basis functions are non-negative. Then, the consistency principle for designing clique functions (in the previous Design Rule) applies to the design of the basis functions. Here, it is sufficient to consider the design of a particular basis function for a single-node clique c, denoted by $B_c(I(R); \mathbf{K}, X(R))$, where, for notational simplicity, the index i has been dropped. According to the consistency requirement between interpretation, measurements and domain knowledge, we want the basis function to be small when $I(R)$, $X(R)$ are *consistent* according to $\mathbf{K}$; otherwise, it should be large. One way to achieve this is to take a probabilistic approach. In particular, we consider the a posteriori pdf $P_c[I(R) \mid \mathbf{K}, X(R)]$. This is the probability that, based on the domain knowledge, $\mathbf{K}$, and the measurement, $X(R)$, the interpretation of the node R should be $I(R)$. By definition, the probability $P_c[I(R) \mid \mathbf{K}, X(R)]$, is such that for $I(R)$ consistent with the measurements and domain knowledge, it is large; otherwise it is small. Hence, a non-increasing function of this pdf can be used as a basis function. For example, the logarithm of the pdf has been suggested [38] as a reasonable basis function for general MRF's. In this case we can define

$$B_c(I(R); \mathbf{K}, X(R)) = -\alpha_c log P_c[I(R) \mid \mathbf{K}, X(R)], \tag{9}$$

where α_c is a positive weighting constant and $-log(\cdot)$ is a monotonically decreasing function. Another way of selecting the basis function is to use

$$B_c(I(R); \mathbf{K}, X(R)]) = \alpha_c(1 - \beta_c P_c[I(R) \mid \mathbf{K}, X(R)]), \tag{10}$$

where α_c and β_c are positive constants, and $\beta_c P_c[I(R) \mid \mathbf{K}, X(R)] < 1$. Usually, we want the normalization constants α_c and β_c to be such that $0 \leq B_c(I(R); \mathbf{K}, X(R)) \leq 1$.

To find the pdf $P_c[I(R) \mid \mathbf{K}, X(R)]$, Bayes's conditional pdf formula can be used. That is,

$$P_c[I(R) \mid \mathbf{K}, X(R)] = P_c[X(R) \mid \mathbf{K}, I(R)] P_c[\mathbf{K}, I(R)] P^{-1}[\mathbf{K}, X(R)], \tag{11}$$

where the first term is the *likelihood functional* of the measurement conditioned on the interpretation, which can be found easily under proper modeling assumptions, and the second term is the prior pdf of the interpretations, which can be determined from a priori information, or heuristically. Finally, the last term is the inverse of the pdf of $X(R)$ which does not depend on I and hence can be dropped in the basis function. For the sake of simplicity, we assume the prior probability to be a constant; that is, the interpretations are equally likely a priori, then the second term can also be dropped.

To further illustrate what we mean by $P_c[I(R) \mid \mathbf{K}, X(R)]$ and how a basis function can be defined from it, consider an example of a single node clique, c. Suppose we have the following *knowledge*:

1. The node could be sky, field, car, road, denoted by interpretation labels L_s, L_f, L_c and L_r.
2. The average gray-level, a feature of the objects above, should be close to G_s, G_f, G_c and G_r respectively, and $G_s > G_r > G_c > G_f$.
3. The distribution of the measured average gray-level of the node, conditioned on each label (sky, field, car, road), is Gaussian. That is, if the measured average gray-level $X(R) = G$, then

$$P[X(R) \mid \mathbf{K}, I(R) = L_\delta] = \frac{1}{\sqrt{2\pi}\sigma_\delta} exp\left[- \frac{(G - G_\delta)^2}{2\sigma_\delta^2} \right], \tag{12}$$

where $\delta = s, f, c, r$. Then, a possible basis function could be from (9)

$$B_c\big(I(R)\,; \mathbf{K}, X(R)\big) = \alpha_c \left(\frac{1}{2} log 2\pi\sigma_\delta^2 + \frac{(G - G_\delta)^2}{2\sigma_\delta^2} \right). \tag{13}$$

Plots of several basis functions, including the one proposed by Modestino [41], are shown in Fig. 7.2.2. Notice that these functions are all "window- like" functions. When domain knowledge about a feature can be expressed in terms of a nominal value, as is in the example above, the specification of the corresponding basis function can be greatly simplified to that of merely constructing one of these window functions. The piecewise linear basis function is particularly interesting in that it is very easy to compute and, as will be seen in later sections, it is relatively robust against measurement errors or image noise. Hence, we give it a special notation, $g(x; a_1, a_2, b_1, b_2)$; where x is the variable and a_1, a_2, b_1, b_2 are the four "corner points", with $a_1 \leq a_2 \leq b_1 \leq b_2$. Similar functions have been used in [22] for a rule-based image interpretation system and in the applications of fuzzy set theory [42].

B.) Clique Functions for Multiple-Node Cliques:

The extension of the clique function design procedure from the case of single-node cliques of *A.*) to the case of multiple-node cliques is quite straightforward. Here, we still design clique functions through designing a set of basis functions, as indicated in expression (7). However, the designing of the basis function is slightly more complicated here in that we may have two types of basis functions. The first type is the basis function for feature measurements, as in the case for single-node cliques. The features in this case could be quantities such as *mutual boundary length, contrast*, etc. Basis functions for these feature measurements can be designed in the same way as that in *A.*) using the window functions of Fig. 7.2.2. The second type of basis functions are those for spatial constraints. The constraints in this case could be statements such as "a car should be on (neighboring to) the road", "a car should never be in the sky", etc. In this case, we can still use the probabilistic approach in the spirit of (9)-(10). For example, consider an arbitrary clique c with multiple nodes denoted by $\mathbf{R}_c$ and interpretations $\mathbf{I}_c(\mathbf{R}_c)$. Let $P_c[\mathbf{I}_c(\mathbf{R}_c) \mid \mathbf{K}]$ be the probability that the combination of interpretations $I_c(\mathbf{R}_c)$ is valid according to domain knowledge. For example, we might have

$$\begin{aligned} P_c[\mathbf{I}_c(\mathbf{R}_c) \mid \mathbf{K}] &= 1; \quad \text{if } \mathbf{I}_c \text{ is a valid combination according to domain knowledge,} \\ &= 0; \quad \text{if } \mathbf{I}_c \text{ is not a valid combination according to domain knowledge.} \end{aligned}$$

Similar to (9)-(10), we can define the basis function as

$$B_c(\mathbf{I}_c(\mathbf{R}); \mathbf{K}) = \alpha_c(1 - P_c[\mathbf{I}_c(\mathbf{R}) \mid \mathbf{K}]). \tag{14}$$

C.) The Selection of the Weights for the Basis Functions:

The weights of the basis functions in (7)-(8) control the contributions of the individual basis functions to the value of a clique function. For simplicity, we may make them all equal. In our current experiments we start with this simple scheme and then, if a feature is too unreliable for a particular object type, we will reduce the corresponding weight. In addition, adjustments are also made by trial-and-error through examining interpretation results on representative training images. A more sophisticated approach, currently under investigation, is to select a weight based on how powerful the corresponding feature is for object recognition and discrimination. For example, consider an arbitrary weight, denoted $p_c^{(i)}(I, \mathbf{K})$, for a single-node clique. It depends on the i'th feature and object label I. If the i'th feature is good for discriminating different objects, or it is a good feature for recognizing object I, the weight $p_c^{(i)}(I, \mathbf{K})$ should be relatively large; otherwise, relatively small. A useful indication of whether a given feature is good for object discrimination can be obtained from the ***inter-cluster distances*** [43], where the clusters are formed by the measurements of the feature from different objects. Similarly, a useful indication of whether a feature is good for recognizing a particular type of object, say type I, can be obtained from the ***intra-cluster standard deviation*** [43], where the cluster is formed by the measurements of the i'th feature on many objects of type I.

D.) Remarks:

To concluue this section, we note several interesting points. First, through the design of clique functions, we have a systematic approach for representing spatial knowledge; that is, for organizing the domain knowledge into a set of well-defined clique functions. This approach also provides guidelines as to what kind of knowledge one would want for the purpose of image interpretation; basically, knowledge concerning objects spatially related as members of different type of cliques. It seems that many of the "rules" in the previous expert systems mentioned in Section 7.2.2 can be transformed into clique functions, where the condition parts correspond to evaluating clique functions and the action parts correspond to assigning labels.

Secondly, under the current neighborhood system assumptions, there are at most four different clique types, as shown in Fig. 7.2.3, which contain at most four nodes. This is due to the fact that the adjacency graph associated with the segmented regions is a *planar* graph, a graph without overpassing edges; while a clique containing five or more nodes causes overpassing of edges in the graph. Hence, the design of clique functions is relatively simple due to the small number of different clique types.

Finally, as has been pointed out in Section 7.2.3, the Gibbs distribution is a very rich distribution in that, as long as the clique functions depend only on the corresponding cliques, their form can be somewhat arbitrary. The general guidelines provided in this section on designing clique functions are based on the considerations of the consistency requirement in the image interpretation problem. While they provide useful insights into the image interpretation problem and offer practical solutions, they are not necessarily the only or the best choice.

7.2.5 Implementation Through Simulated Annealing:

In the last section, image interpretation is formulated as a MAP estimation problem. Under the MRF modeling assumption, this becomes the problem of minimizing a properly defined energy function such that the interpretation obtained is most consistent with measurements and domain knowledge. The simplest optimization method is an exhaustive search procedure. This, however, results in an exponential complexity of $O(M^N)$, where M is the number of labels and N is the number of nodes in the adjacency graph. An alternative is the simulated annealing algorithm, a stochastic iterative optimization procedure, that will find the global maximum of the pdf of the MRF, or the minimum of the energy function, without excessive computation [29],[48]. The simulated annealing algorithm has been widely used in various applications involving combinatorial optimization, such as VLSI layout [44], channel coding [45], image segmentation [46]-[47].

For convenience, we rewrite this algorithm here in the context of a minimization problem. Let the function to be minimized be $E(\mathbf{x})$, where $\mathbf{x}$ is the indepdendent variable. This algorithm can be loosely described as follows:

Simulated Annealing

1) Select an initial "temperature" parameter T_0 and randomly choose an initial variable $\mathbf{x}_0$. Iteration begins.

2) At step k, perturb $\mathbf{x}_k$ by $\hat{\mathbf{x}}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}$ and compute $\Delta E = E(\hat{\mathbf{x}}_{k+1}) - E(\mathbf{x}_k)$.

3) If $\Delta E < 0$ accept the change; that is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta\mathbf{x}.$$

If $\Delta E > 0$, accept the change only with probability $p = e^{-\Delta E/T}$.

4) If there is a considerable drop in energy, or enough iterations, lower the temperature.

5) If the energy becomes stable and the temperature is very low, stop; otherwise go back to (2).

For image interpretation, the implementation is straightforward when the definition of the method of perturbation and the annealing schedule (i.e., how the temperature is lowered) are decided. We first order all the nodes arbitrarily as node $1, 2, ..., N$. Then, an iteration is defined as *one* visit to *all* the nodes according to this order. When a node is visited, a perturbation of the interpretation vector is performed through generating a new label for this node from an uniform distribution of all the possible interpretations, $L_1, L_2, \ldots, L_M$, where M is the number of different labels, or from the conditional probability distribution of the MRF (i.e., the Gibbs sampler of [29]). In our experiments, we found that the two perturbation methods provide the same results in terms of converging to the optimal interpretations, while the Gibbs sampler is more complicated in computation structure, so we have mainly made use of the first approach for perturbation. Finally, for the annealing schedule, the temperature is lowered after each iteration according to $T_{l+1} = \alpha T_l$ where $0.5 < \alpha < 1$, which we have used successfully in an MRF model-based MAP image segmentation procedure [46]. In particular, we have selected $T_0 = 1$ and $\alpha = 0.92$ for all our experiments.

7.2.6 Interpretation of Synthetic Images:

After the discussion of the previous sections, it may be expected that the efficacy of the MRF model-based approach depends on several factors, including the validity of the MRF assumption, the quality of the segmentation, how powerful the features and spatial constraints in the knowledge base are (as far as object recognition is concern), and finally, how effective the simulated annealing is (convergence of interpretations). Before applying the MRF approach to real-world image interpretation, then, it make sense to test whether this approach would work at all under the somewhat ideal situation in which we have acceptable segmentations and relatively strong features and spatial constraints in the domain knowledge. If the MRF model-based approach works well here, then it is reasonable to expect that it will be effective for real-world image interpretation provided we can produce good segmentations (or are able to deal with poor ones) and find strong features and spatial constraints. We can create such an ideal situation through generating synthetic images and studying the performance of the MRF model-based approach in interpreting them. In this section, we describe some of the experimental results on synthetic images taken from a more complete study [41],[46].

The synthetic images used in this experiment are variations from the conceptual image of Fig. 7.2.1 which contains such objects as sky, road, field, and car, all of which appear as regions of *constant* gray-levels. The assumed domain knowledge associated with this image is shown in Table 7.2.1 where object features (with precise definitions in Table 7.2.2) and spatial constraints are stated. In the experiments [41],[46], we have found that, compared to other basis functions of Fig. 7.2.2, the piecewise-linear basis functions were more effective for object recognition and less sensitive to segmentation error. Hence, all the results for image interpretation in this paper have been obtained using clique functions composed of this type of basis function. In Table 7.2.3, these clique functions are shown in terms of their basis functions and weights for the synthetic image (corner points a_1, a_2, b_1, b_2, weight p, and α_c for the basis function for spatial constraints). Finally, the segmentation algorithm used here is a Gaussian model-based segmentation algorithm [49] which has been quite effective for aerial photograph segmentation.

The experiments on the synthetic image described in this section contain three parts. First, the "ideal image" of Fig. 7.2.1 is interpreted with result shown in Fig 7.2.4. In this case, the extracted segmentation is perfect, as shown in Fig. 7.2.4 b, since all the regions have constant gray levels. The interpretation result in Fig. 7.2.4 c, in which different gray- levels indicate different object labels (as shown in Fig. 7.2.4 d), shows that all the regions are correctly identified. This suggests that when the image is well-segmented and the domain knowledge is sufficient, the MRF model-based approach is effective. As reported in [46], starting from a random initial interpretation vector, the simulated annealing converged within 25 iterations. In this, and the rest of the results of this section, the clique functions of Table 7.2.3 have been used, and the number of iteration for the simulated annealing is set to 25.

In the second part of the experiments, the ideal image of Fig. 7.2.1 is corrupted by additive white Gaussian noise to generate degraded images of different signal-to-noise ratios (SNR's). These images, then, are presented for interpretation. Here, the added noise should result in errors in segmentation and feature measurements. This is used to study the performance of the MRF approach under moderately imperfect segmentation. In [46] experiments have been performed for images with SNR of 20dB, 10dB, and 3dB with similar results. Hence, the results are only shown in Fig. 7.2.5 for the 3dB case. Again, all the objects are correctly identified. Here, the car has very a small area and its identification might be most seriously affected by segmentation error. However, since the road can be well identified, the car can still be identified partly due to the spatial constraints between them. To be able to deal with more serious segmentation errors, such as the case in which the car is split into several regions, the clique functions have to be expanded to allow the merging of regions. This is currently under investigation [27].

In the last part of the experiments, we have consider the case where there are *unknown* objects in the image for which no information is available in the knowledge base. There are two main causes for unknown regions, e.g., the region belongs to an unknown object or the region belongs to a known object but the feature measurements are far from the nominal values of all known objects due to segmentation errors. In both cases, it is more desirable to assign an "unknown" label to such regions than to risk making a mistake. Hence, we have added an unknown label to the set of object labels. In this part, an "unknown" object which is similar to a car is placed in the right "field" region of the images used in the preceding experiments and the resulting image has been re-interpreted [46]. Here, we have only shown the results for the ideal image and the image with 3dB SNR in Fig.'s 7.2.6 and 7.2.7, respectively. All the regions have been identified correctly and the unknown region is not identified as a car since that will violate the spatial constraint in the knowledge base. This example, to some extent, shows the flexibility of the MRF approach.

7.2.7 Interpretation of Real-World Images:

To test the practical applicability of the proposed MRF approach for image interpretation, experiments have been performed on real-world images; in particular, aerial photographs, which have been digitized to 256 gray-level images of 256 × 256 pixels. Experimental results obtained here also provide further insights and useful guidelines on how to effectively apply this general approach in practice. Since we have no control over the generation process of the images involved, the task of interpreting real-world images is much more complicated and difficult than that of interpreting synthetic images. We have proceeded in two steps; namely, knowledge acquisition through constructing clique functions of the MRF model and interpretation using simulated annealing.

A.) Knowledge Acquisition:

This is the process of gathering information about the objects of interest in real-world images. This information is usually represented in terms of features and spatial constraints. For example, we might like to obtain information about cars, such as "a car has an average area of 800 pixels" and "a car is always on (neighboring to) a road". Here, the area is a feature while the neighborhood relationship is a spatial constraint. Knowledge acquisition is also a *selection* process in which we select certain features and constraints from *all* the features and constraints we know about the objects to form a knowledge base. The selection process is necessary, since some of the features and constraints are not essential to interpretation, while they only add system complexity and computational burden. In the selection of the features and constraints, we want to select the ones that are powerful for object recognition and discrimination. At this point, this selection is performed heuristically through trial-and-error. As a future goal, a general approach to solve this problem, such as the one described in Section 7.2.4 C, needs to be found.

There are many sources for knowledge acquisition. For aerial photointerpretation, one source of information/knowledge is from map information and characteristics of the man-made objects in the area being photographed [20]. This approach is used often in constructing practical photointerpretation systems and it usually involves a large amount of information. A less complicated alternative is the training approach. In this approach, a number of representative or training images are first segmented and interpreted by human experts; knowledge is then extracted from these segmented and interpreted images. The training approach is relatively simple, while not sacrificing generality in principle; hence is very useful in experimental studies such as this. In this work, we make exclusive use of the training approach.

For simplicity, only one training image is used in this experiment which is an aerial photograph as shown in Fig. 7.2.8. After human expert segmentation and interpretation performed using an interactive display-segmentation facility at RPI [51], it has been found to contain mainly the following types of objects: 1.)vegetation region (VEGE); 2.)shadow (SHD1); 3.)shadow on the ground (SHD2); 4.)ground (GRND); and 5.)oil tank (OLTK). The feature measurements selected for the objects includes: area, average gray-level, compactness of an object, and contrast between two regions (see the definitions in Table 7.2.2). The constraints selected for the objects include a number of neighboring relationships. For the clique functions, we have used again the piecewise-linear basis functions for the features and basis functions of (14) for spatial constraints. In particular, the corner points of the basis function for a given feature of a given object is determined from observing the *maximum* and *minimum* of the measurements of that feature on this type of object identified by the human interpreter. "Guard intervals" are introduced around the the maximum and minimum to determine the exact values of the four corner points. This, as well as the selection of the weights, has been performed heuristically, since there are relatively few objects of each type. When the number of objects of different types is large, the process of determining the corner points can be automated [50]. In Table 7.2.4, we have shown the domain knowledge learned from the training data in the form of the basis functions from which the clique functions are constructed. This set of basis functions and subsequent clique functions have been used to obtain all the experimental results to be described in this section.

B.) Interpretation Using Simulated Annealing:

In this experiment, interpretation has been performed on two test images. The first one is the training image itself. This is used to verify the correctness and effectiveness of the knowledge obtained in the form of clique functions from the training stage. Here we have performed interpretation on the training image, using both manual segmentation and computer segmentation, as shown in Fig.'s 7.2.8 and 7.2.9, respectively. Since the computer segmentation provides comparable quality to that of the manual segmentation, the subsequent interpretation results are both quite good. Here, most of the regions are correctly identified except that some of the oil tanks, which appear only partly in the image, are labeled as unknown objects. The reason for this is that in the knowledge base, oil tanks are characterized as circular objects, as reflected from the definition of the compactness in Table 7.2.2 and the corner points for the corresponding basis functions in Table 7.2.4. In other words, oil tanks that are only partly in the image were treated as unknown objects in the training stage and hence, it is not surprising that they have been interpreted as so in the interpretation stage. To recognize these "partial" oil tanks, more powerful shape features should be used, as described below.

The second test image, as shown in Fig. 7.2.10 a, is "cut" from a larger image (2048 × 2048), from which the training image is obtained, and contains similar objects to those in the training image. This image is used to test the usefulness of the knowledge and clique functions obtained from the training image. Notice that in the original image the gray tone and texture of several oil tanks are so close to those of their surroundings that they are very hard to extract even by human eyes; it is then not surprising that the computer segmentation of several regions corresponding to oil tanks is rather poor, as shown in Fig. 7.2.9 c, especially in their shapes. As a result, in the interpretation results shown in Fig. 7.2.10 c, most of the regions are correctly identified except for these oil tank regions. In fact, the compactness feature failed to be effective, since the regions corresponding to the oil tanks have very noisy boundaries and some of them are only partially in view. To solve this problem, Yu has proposed a more robust shape feature, called partial compactness, for the oil tanks [52]. This feature is based on the idea of obtaining a large number of estimates of the radius of a segmented region from random points on the boundary of the region, as illustrated in Fig. 7.2.11. Here, we have shown a set of three random points, A, B, and C, on the boundary of a segmented region. The vertical equal division lines of AB and BC intersect at O, resulting in a random estimate of the radius r. In this way, every set of three random boundary points provides a random estimate of radius. If a region is reasonably circular, or partially circular, the variance of the random estimates tends to be small. It has been shown by Yu [51] that this feature is quite powerful for recognizing both circular and partially circular objects, and is relatively robust to noisy boundaries. An interpretation of the second image is performed, again, using the knowledge-base and clique functions of Table 7.2.4, except with the compactness replaced by Yu's partial compactness (also shown in Table 7.2.2 and 7.2.4). As can be seen from the interpretation results shown in Fig. 7.2.12 c, improvements are obtained in that all the oil tanks are correctly identified. This justifies the points made in the experiments on interpreting synthetic images, that the MRF model-based formulation is quite powerful and feature selection is the crucial problem in applying it to real-world image interpretation.

C.) Remarks:

In concluding this section, we want to point out that purpose of the set of simple experiments performed here is to understand some of the fundamental problems in knowledge-based image interpretation, such as knowledge representation, the selection of features and constraints, and the interaction between interpretations of different regions. While this should provide useful insights into how to build practical systems, the knowledge base and clique functions used here are not sufficient yet as a practical system. For example, the features used in these experiments are neither meant to be sufficient nor the best to use, as would have been done in building practical systems; but rather, they are used here to demonstrate useful concepts and important issues related to the performance of knowledge-based image interpretation systems. Finally, a practical system should also incorporate the capability of recognizing complex objects composed of simpler ones; for example, recognizing a runway area from road-like regions containing a number of airplanes. It appears that the MRF model-based formulation can be use here to construct a hierarchical representation for objects of different complexity. In this representation, the regions corresponding to simple objects form a low-level MRF, while regions corresponding to complex objects which are collections of simple object regions form a high-level MRF. The optimal interpretation is the one that maximizes the pdf of this hierarchical MRF. In fact, a similar hierarchical MRF model has been used successfully for image segmentation using the pyramid image structures [53].

7.2.8 Summary and Future Research:

In this paper, we have described an MRF model-based approach for automated image interpretation demonstrated as a region-based approach. In this approach, an image is first segmented into a collection of disjoint regions of certain homogeneous image properties. These regions, together with their spatial relationships, form an adjacency graph, and image interpretation is achieved through assigning object labels, or interpretations, to the regions using domain knowledge and measurements of features and spatial relationships extracted from them. In this approach, the interpretations are modeled as a MRF on the adjacency graph and the image interpretation problem is formulated as that of finding the *best* realization of the MRF given the domain knowledge and measurements. Through the MRF model, this approach provides a systematic methodology for organizing and representing domain knowledge through properly designed clique functions associated with the pdf of the underlying MRF, which are to be designed in such a way that the optimal interpretation found is most consistent with domain knowledge and measurements. In particular, we have proposed a structure for the clique functions as a weighted sum of basis functions of features and spatial constraints. Finally, the simulated annealing algorithm is used to find the globally optimal interpretation.

To study the efficacy of the MRF model-based approach, image interpretation experiments have been performed on both synthetic images and real-world aerial photographs. In the experiments, we have found the piecewise-linear basis function provides robust performance for the interpretation in the presence of measurement errors caused by imperfect segmentation. We have also found that selecting powerful features and constraints are very crucial to real-world image interpretation; when such features and constraints are used, most of the objects in the image are correctly recognized.

Although the results here are still preliminary, they do suggest several a promising directions for future research work. Specifically, future research should include the following three immediate research tasks. First of all, a more general approach is needed to determine the weights and corner points for the piecewise-linear basis functions used to construct of the clique functions. Secondly, work should be done to incorporate other primitives, such as linear edge segments, and high-level to low-level feedback, such as the split-and-merge of original segmented regions during interpretation, into the MRF model-based approach, as described in Section II. Some preliminary results for these two task have already been obtained [27]-[28]. Finally, the MRF model-based approach needs to be tested on more diverse real-world images, such as additional aerial photographs.

References for Section 7.2

1. T. E. Avery, *Interpretation of Aerial Photographs*, Burgess Publishing Company, Minneapolis, Minnesota, 1977.
2. S. A. Drury, *Image Interpretation in Geology*, Allen and Unwin, Ltd., London, UK, 1987.
3. D. P. Paine, *Aerial Photography and Image Interpretation for Resource Management*, John Wiley and Sons, Inc., New York, 1981.
4. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.
5. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, 2nd Ed., Academic Press, New York, 1982.
6. B. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.
7. Proc. of the 6th Int. Conf. on Robot Vision and Sensor Control Paris, France, 1986.
8. R. A. Schowengerdt, *Techniques for Image Processing and Classification in Remote Sensing*, Academic Press, New York, 1983.
9. N. J. Nilsson, *Principles of Artificial Intelligence*, Tioga Pub. Company, Palo Alto, CA, 1980.
10. R. D. Keller, *Expert System Technology: Development and Application*, Lourdon Press, Englewood Cliffs, NJ, 1987.
11. M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*, Plenum Press, New York, 1980.
12. T. Binford, "Survey of model-based image analysis systems", The Int. J. of Robotics Research, Vol. 1, No. 1, pp. 587-633, 1982.
13. Y. Ohta, *Knowledge-based Interpretation of Outdoor Natural Color Scenes*, Pitman Advanced Publishing Program, Boston, 1985.
14. O. D. Faugeras and K. E. Price, "Semantic description of aerial images using stochastic relaxation", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-3, pp. 633-642, Nov., 1981.
15. K. E. Price, "Relaxation matching techniques: a comparision", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-7, pp. 617-623, Sept., 1985.

16. V. S. S. Hwang, "Evidence accumulation for spatial reasoning in aerial image understanding", Ph.D. Thesis, Univ. of Maryland, College Park, 1984.

17. V. S. S. Hwang, T. Matsuyama, L. Davis, and A. Rosenfeld, "Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding", CS-TR-1300, Univ. of Md., 1983.

18. R. Prasannappa, L. Davis, and V. S. S. Hwang, "A knowledge-based vision system for aerial image understanding", CS-TR-1758, Univ. of Md., Jan., 1987.

19. R. Brooks, "Symbolic reasoning among 3-dimensional models and 2-dimensional images", *Artificial Intelligence*, Vol. 17, pp. 285-394, 1981.

20. D. M. McKeown, W. A. Harvey, and J. McDermott, "Rule-based interpretation of aerial imagery", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-7, pp. 570-585, Sept., 1985.

21. D. M. McKeown and W. A. Harvey, "Automating knowledge acquisition for aerial image interpretation", TR., CMU-CS-87-102, Carnegie-Mellon Univ., Jan., 1987.

22. E. M. Riseman and A. R. Hanson, "A methodology for the development of general kowledge-based vision systems", TR., COINS 86-27, Univ. of Mass., July, 1986.

23. A. R. Hanson and E. M. Riseman, "From image measurements to object hypothesis", TR., COINS 87-129, Univ. Mass, Dec., 1987.

24. B. Draper, R. Collins, J. Brolio, A. R. Hanson and E. M. Riseman, "The schema system", TR., COINS 88-76, Univ. Mass, Sept., 1988.

25. E. M. Riseman and A. R. Hanson, "Summary of image understanding research at the University of Massachusetts", TR., COINS 88-32, Univ. of Mass., May, 1988.

26. "Technical reports of the computer vision laboratory, 1986-1988", Univ. of Md., 1988.

27. J. Zhang, "Merging of segmented regions through simulated annealing", RPI Technical Report, in preparation.

28. J. Zhang, "Consistent Combination of local interpretation for image analysis", RPI Technical Memo, May, 1988.

29. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images," IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-6, pp. 721-741, Nov., 1984.

30. C. W. Therrien, T. F. Quatieri and D. E. Dudgeon, "Statistical model-based algorithms for image analysis", Proc. IEEE, Vol. 74, April, 1986.

31. J. Zhang and J. W. Modestino, "Markov random fields with applications to texture classification and discrimination", Proc. The 20th Annual Conf. on Information Science and Systems, Princetion University, NJ, March, 1986.

32. H. Derin and H. Elliot, "Modelling and segmentation of noisy and textured images using Gibbs random fields", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 39-55, Jan., 1987.

33. F. S. Cohen and D. B. Cooper, "Simple, parallel, hierachical and relaxation algorithms for segmenting non-casual Markovian random field models", Proc. IEEE Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 195-219, March, 1987.

34. J. Besag, "On the statistical analysis of dirty pictures", J. Royal Stat. Soc. B., Vol. 48, pp. 259-302, 1986.

35. J. Zhang and J. W. Modestino, "Unsupervised image segmentation Using a Gaussian model", to be submitted to IEEE Trans. PAMI.

36. J. Marroquin, S. Mitter, and T. Poggio, "Computer vision", J. Amer. Stat. Association, Vol. 82, pp. 76-89, March, 1987.

37. P. B. Chou and C. M. Brown, "Multi-modal segmentation using Markov random fields", Proc. IJCAI, pp. 663-670, 1987.

38. R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications*, Providence, RI: Amer. Math. Soc., 1980.

39. J. Besag, "Spatial interaction and the statistical analysis of lattice systems", J. Roy. Statist. Soc., Series B., Vol. 36, pp. 192-226., 1974.

40. J. A. Feldman and Y. Yakimovsky, "Decision theory and artificial intelligence: I. a semantic-based region analyzer", Artificial Intelligence, Vol. 5, pp. 325-348, 1974.

41 J. W. Modestino, "A hierarchical region-based approach to automated photointerpretation", RPI report, March 1987.

42. L. Zadeh, "Approximate Reasoning Based on Fuzzy Logic", Proc. 6th IJCAI, pp.1004-1010, 1979.

43. J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesly Pub.

Company, Reading, MA, 1974.

44. S. Kirkpatrick, C. S. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing", *Science*, Vol. 220, pp. 671-680, May, 1983.

45. A. A. El Gamal, L. A. Hemachandra, I. Shperling and V. K. Wei, "Using simulated annealing to design good codes", IEEE Trans. Infor. Theory, Vol. IT-33, pp. 116-123, Jan., 1987.

46. J. Zhang, "Two-dimensional stochastic model-based image analysis", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, Aug., 1988.

47. C. S. Won and H. Derin, "Segmentation of noisy textured images using simulated annealing", Proc. ICASSP, pp. 563-566, 1987.

48. B. Gidas, "Nonstationary Markov chains and convergence of the annealing algorithm", J. Statist. Phys., Vol. 39, pp. 73-131, 1985.

49. J. Zhang and J. W. Modestino, "Image segmentation using a Gaussian model", Proc. Conf. on Info. Sci. and System, Princeton University, NJ, March, 1988.

50. J. Zhang, "Learning in the MRF model-based approach for image interpretation", in preparation.

51. J. Kanai, "Interpretation of real images using the MRF model- based method", RPI report, September, 1987.

52. L. Y. Yu, "New results on image segmentation and interpretation using the MRF model", RPI report, June, 1988.

53. C. Bouman and B. Liu, "Segmentation of textured images using a multiple resolution approach", Proc. ICASSP, pp. 1124-1127, New York, March, 1988.

54. A. Rosenfeld, R.A. Hummel, and S.W. Zucker, "Scene labeling by relaxation operations," IEEE Trans. Syst., Man, Cybern., Vol. SMC-6, pp. 420- 433, 1976.

55. R.M. Haralick and L.G. Shapiro, "The consistent labeling problem: Part I," IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-1, pp. 173-184, Apr. 1979.

56. R.M. Haralick and L.G. Shapiro, "The consistent labeling problem: Part II," IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-2, pp. 193-203, May 1980.

57. R.A. Hummel and S.W. Zucker, "On the foundation of relaxation labeling process," IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-5, pp. 267-287, May 1983.

Table 7.2.1 ·Summary of Assumed Knowledge for the Conceptual Image

a. a single region

	Region Knowledge	
Object Type	Area (No. of pixels)	Average Gray Level
Car	≤ 800	≐ 150
Sky	≥ 25000	≐ 200
Road	≐ 11700	≐ 100
Field	≥ 13500	≐ 50

b. two regions

			Mutual Knowledge
Object Type	Boundary Length	Contrast	Spatial Constraints
Sky, Car	0	—	impossible combination
Field, Car	0	—	impossible combination
Sky, Road	≐ 56	≐ 100	valid combination
Car, Road	≐ 120	≐ 50	valid combination
Sky, Field	≐ 100	≐ 150	valid combination
Road, Field	≐ 180	≐ 150	valid combination

c. three regions

	High - Order Knowledge
Object Types	Spatial Constraints
Sky, car, field	impossible combination
Sky, car, road	impossible combination
Sky, road, field	valid combination
Road, car, field	impossible combination

Table 7.2.2 **Definitions of Several Region-Based Features**

a.) Features for a Single Region R

1. Area:

$$A = \text{the number of pixels in the region } R.$$

2. Average Gray Level:

$$G = \frac{1}{A} \sum_{(i,j) \in R} x(i,j),$$

where $x(i,j)$'s are gray levels of the pixels in R.

3. Standard Deviation of Gray Levels:

$$S = \left(\frac{1}{A} \sum_{(i,j) \in R} \left(x(i,j) - G \right)^2 \right)^{1/2}.$$

4. Compactness (Kanai's):

$$C = \frac{P^2}{4\pi A} - 1,$$

where P is the perimeter of R

$$P = \text{the number of boundary points of } R.$$

5. Partial Compactness (Yu's):

$$C_p = \text{sample standard deviation of } r,$$

where r is the random measurement of the radius of region R as shown in Fig. 5.20.

b.) Features for Two Adjacent Regions R_i and R_j

1. Boundary Length:

$$B_{i,j} = \frac{P_i + P_j}{2}.$$

2. Contrast:

$$C_{i,j} = |G_i - G_j|.$$

Table 7.2.3 Linear Basis Functions for the Synthetic Image.

a. cliques of a single node

	Features	
	Area	Average Gray Level
Label	a1, a2, b1, b2, p	a1, a2, b1, b2, p
car	750,790,810,850,0.5	165,175,185,187,0.5
road	11500,11600,11900,12000,0.5	85,95,105,115,0.5
sky	19900,20000,65536,65536,0.5	193,195,205,215,0.5
field	13300,13400,13900,14000,0.5	0,0,55,65,0.5

b. cliques of two nodes

	Features		
	Boundary length	Contrast	Spatial Constraints
Labels	a1, a2, b1, b2, p	a1, a2, b1, b2, p	αc, p
sky, car			1.0, 1.0
field, car			1.0, 1.0
sky, road	0,0,55,65,0.5	85,95,105,115,0.5	0.0, 1.0
car, road	105,115,125,135,0.5	0,0,85,95,0.5	0.0, 1.0
road, field	140,150,1000,1000,0.5	35,45,55,65,0.5	0.0, 1.0
sky, field	85,95,105,115,0.5	140,150,257,257,0.5	0.0, 1.0
all others	—	—	0.5, 1.0

c. three regions

Labels	αc, p
sky, car, field	1.0, 1.0
sky, car, road	1.0, 1.0
sky, road, field	0.0, 1.0
road, car, field	1.0, 1.0
all others	0.5, 1.0

Table 7.2.4 Knowledge and Clique Functions for the Aerial Photos

a basis function for cliques of a single node

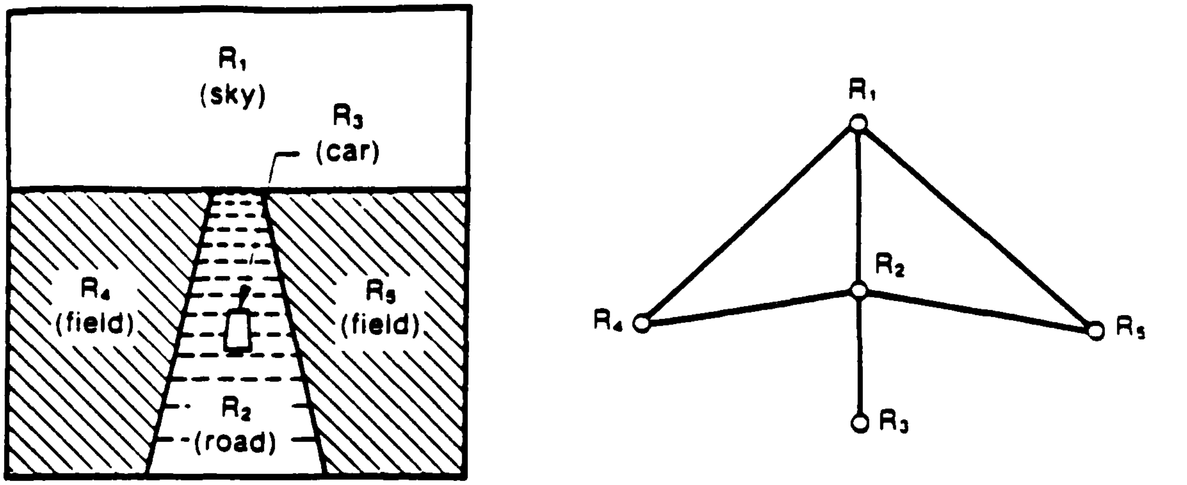
	Features	
	Area	Average Gray Level
Label	a1. a2. b1. b2. p	a1. a2. b1. b2. p
vegetation	11500.12000.13000.13500.0.5	120.123.127.130.0.5
shadow 1	-	135.140.150.155.1.0
shadow 2	0.0.6500.7000.0.5.	105.110.140.145.0.5
ground	-	185.190.200.205.1.0
oil tank	4500.4500.64000.64000	200.205.256.256.0.33

a. basis function for cliques of a single node (cont.)

	Features	
	Compactness	Partial Compactness
Label	a1. a2. b1. b2. p	a1. a2. b1. b2. p
vegetation	-	-
shadow 1	-	-
shadow 2	-	-
ground	-	-
oil tank	0.0.0.0.0.95.0.95. 1.0	0.0.0.0.10.0.10.0.1.0

b. basis function for features and spatial constraints
for cliques of two nodes

	Feature	Spatial Constraint
	Contrast	
Lables	a1. a2. b1. b2. p	αc. p
vegetation and shadow 1	10.15.20.30.1.0	0.0. 1.0
shadow1 and shadow 2	0.5.30.35.1.0	0.0. 1.0
shadow 1 and ground	35.40.60.65.1.0	0.0. 1.0
shadow 1 and oil tank	45.50.75.80.1.0	0.0. 1.0
shadow 2 and ground	55.60.90.95.1.0	0.0. 1.0
shadow 2 and oil tank	75.80.105.110.1.0	0.0. 1.0
ground and oil tank	0.5.25.30.1.0	0.0. 1.0
vegetation and oil tank	impossible combination	1.0. 1.0
oil tank and oil tank	impossible combination	1.0. 1.0
all other cases		0.5. 1.0



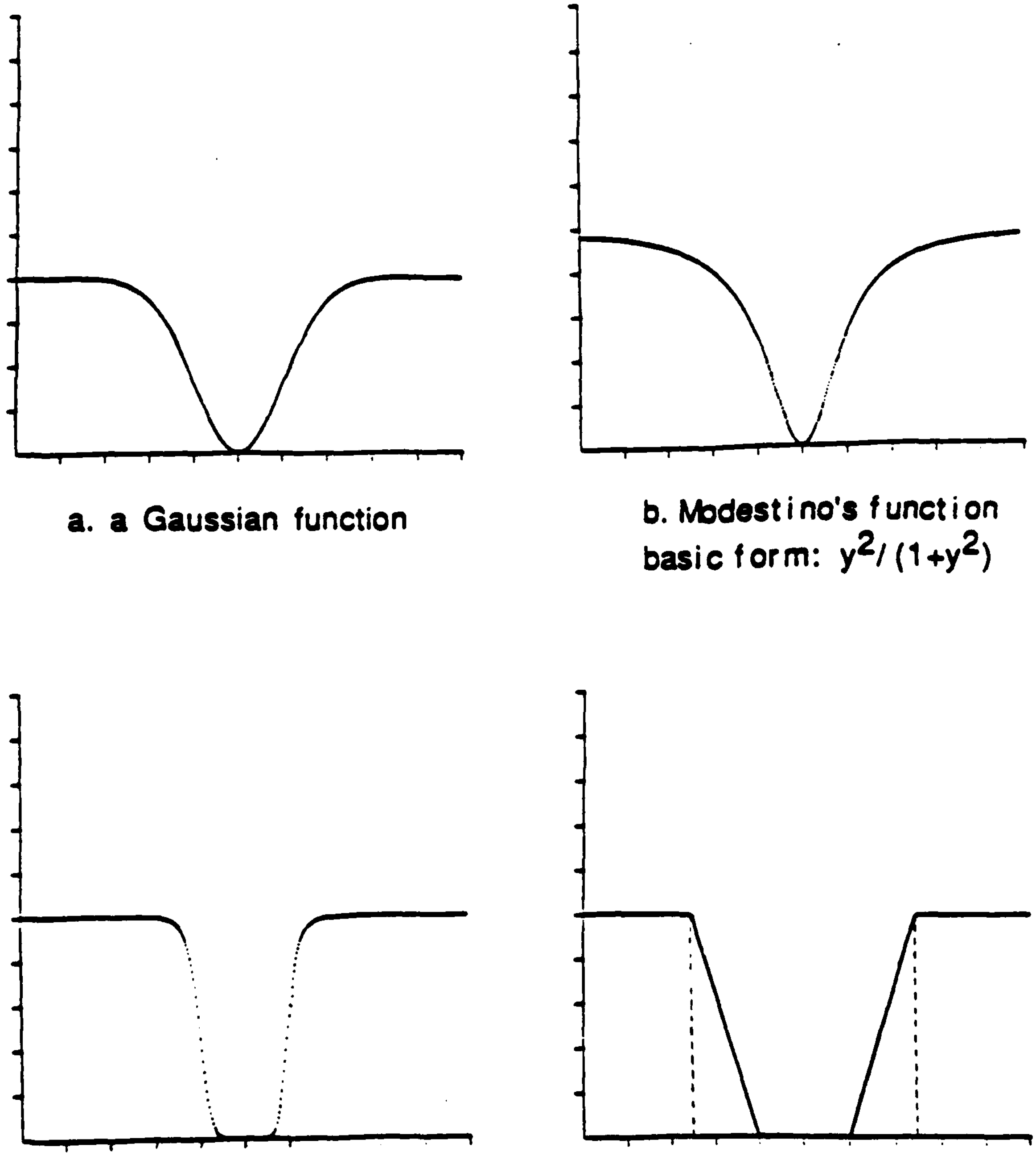
a. The synthetic conceptual image

b. the adjacency graph of the conceptual image

Node or Region	Associated Cliques
R_1	$\{R_1\}, \{R_1, R_2\}, \{R_1, R_4\}, \{R_1, R_5\},$ $\{R_1, R_2, R_4\}, \{R_1, R_2, R_5\}$
R_2	$\{R_2\}, \{R_2, R_3\}, \{R_2, R_4\}, \{R_2, R_5\}$
R_3	$\{R_3\}$
R_4	$\{R_4\}$

c. cliques of the adjacency graph

Figure 7.2.1 A Synthetic Conceptual Image for Image Interpretation.



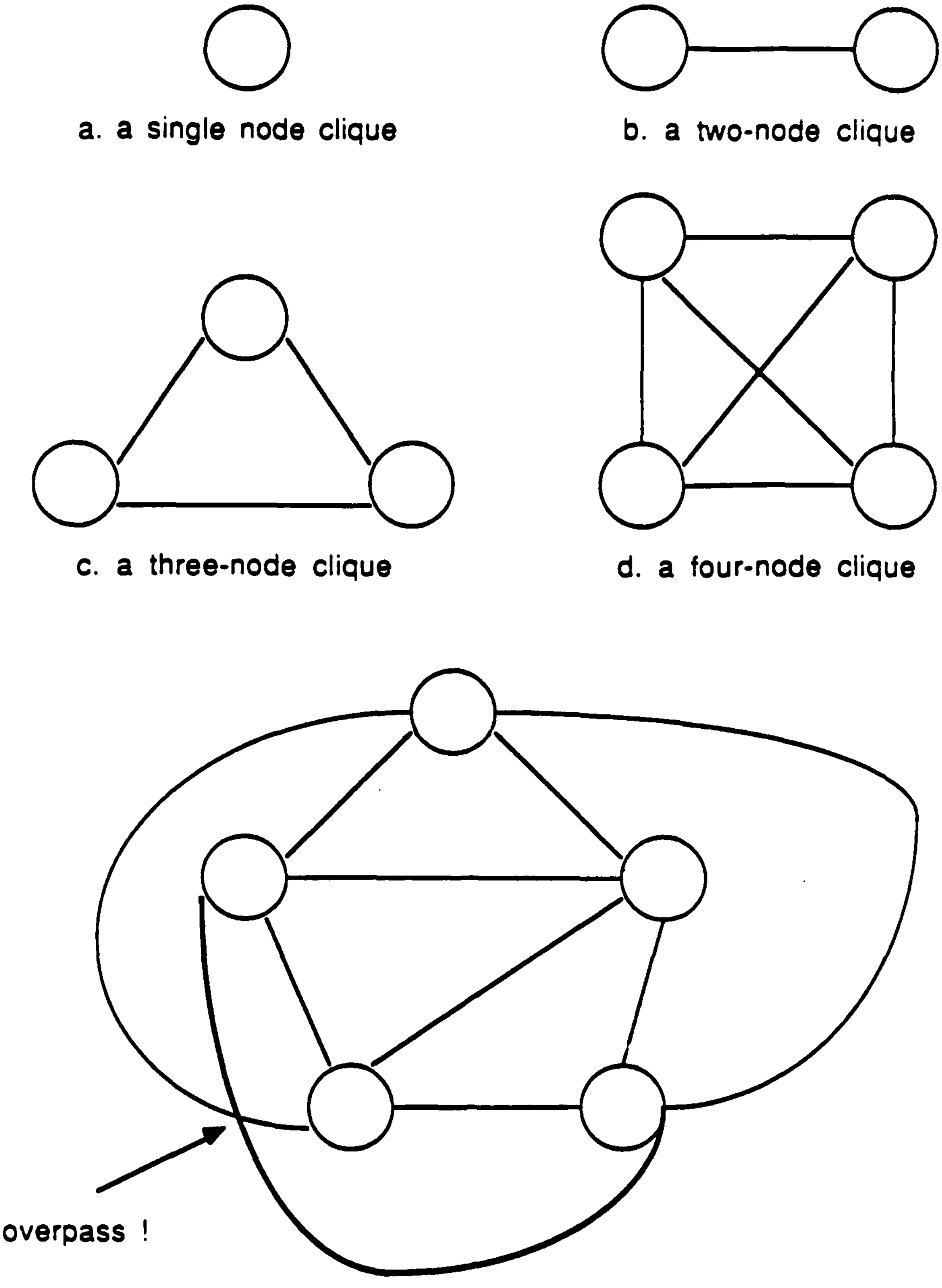
a. a Gaussian function

b. Modestino's function
basic form: $y^2/(1+y^2)$

c. extension to Modestino's function
basic form: $y^n/(1+y^n)$, n=8

d. the piecewise linear function

Figure 7.2.2 Example of Different Basis Functions.



a. a single node clique

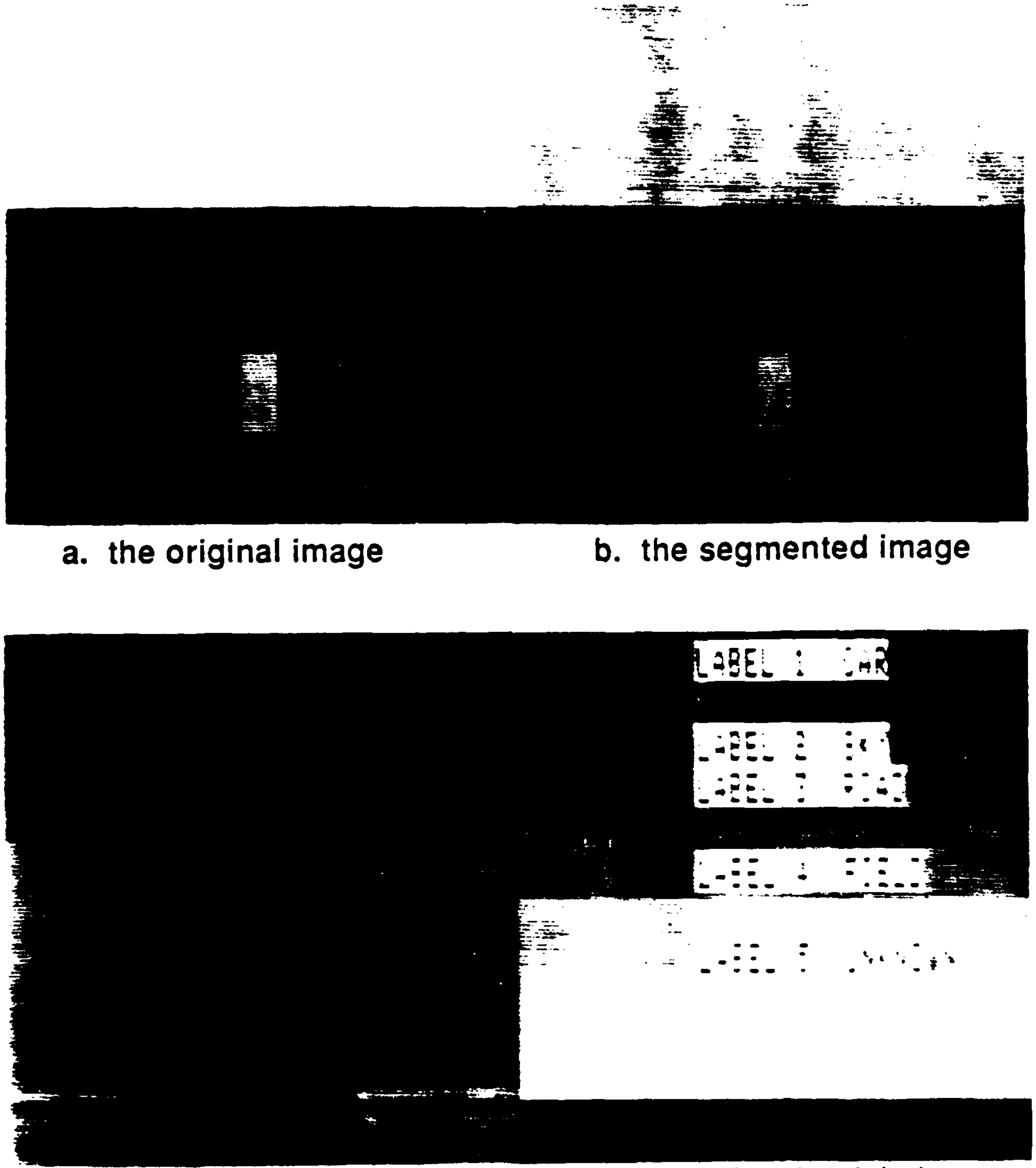
b. a two-node clique

c. a three-node clique

d. a four-node clique

e. five nodes can not form a clique

Figure 7.2.3 Illustration of All Possible Cliques for a First-Order Neighborhood System.



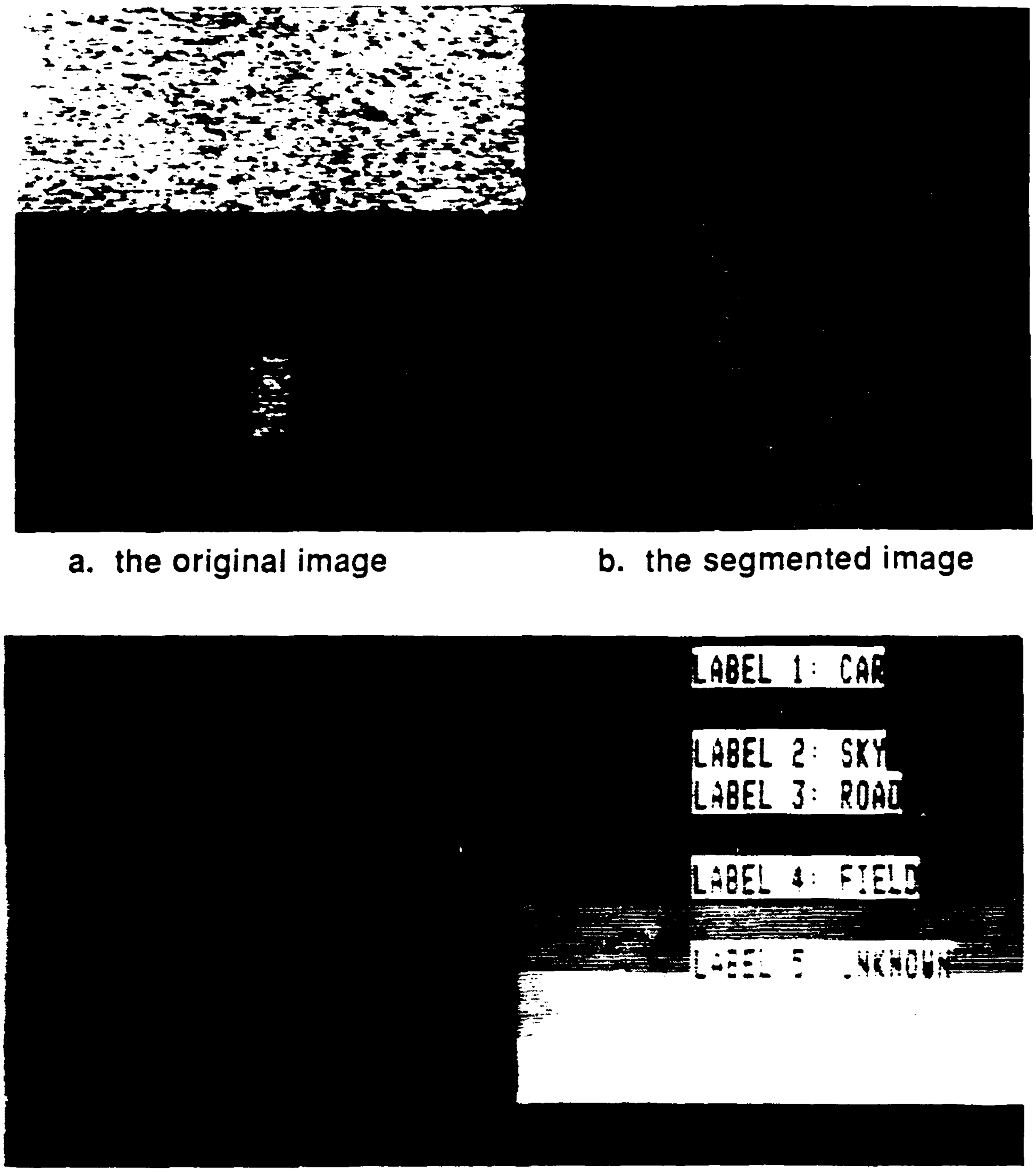
a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

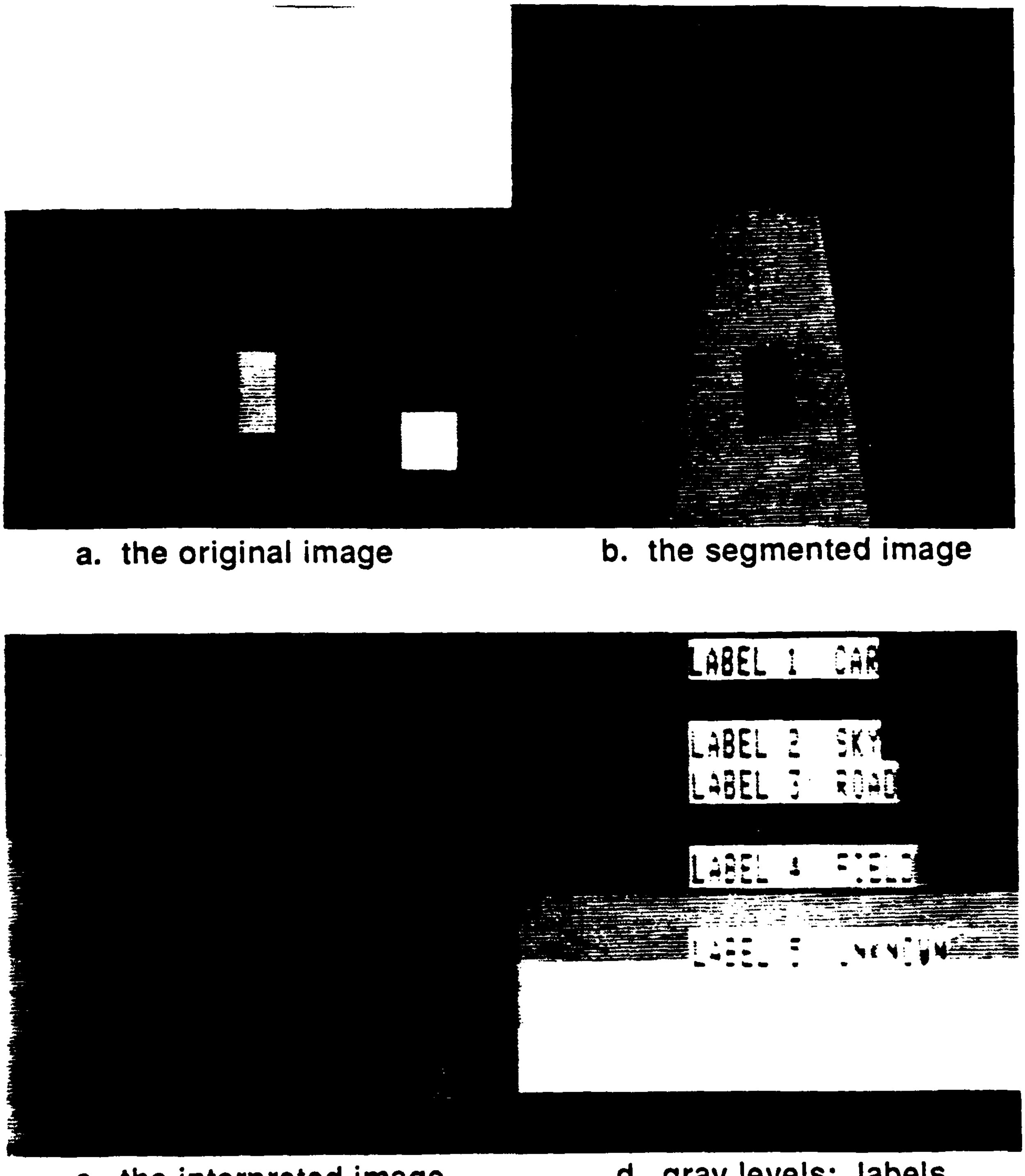
Figure 7.2.4 Interpretation of the Ideal Image



a. the original image b. the segmented image

c. the interpreted image d. gray levels: labels

Figure 7.2.5 Interpretation of the 3dB SNR Image



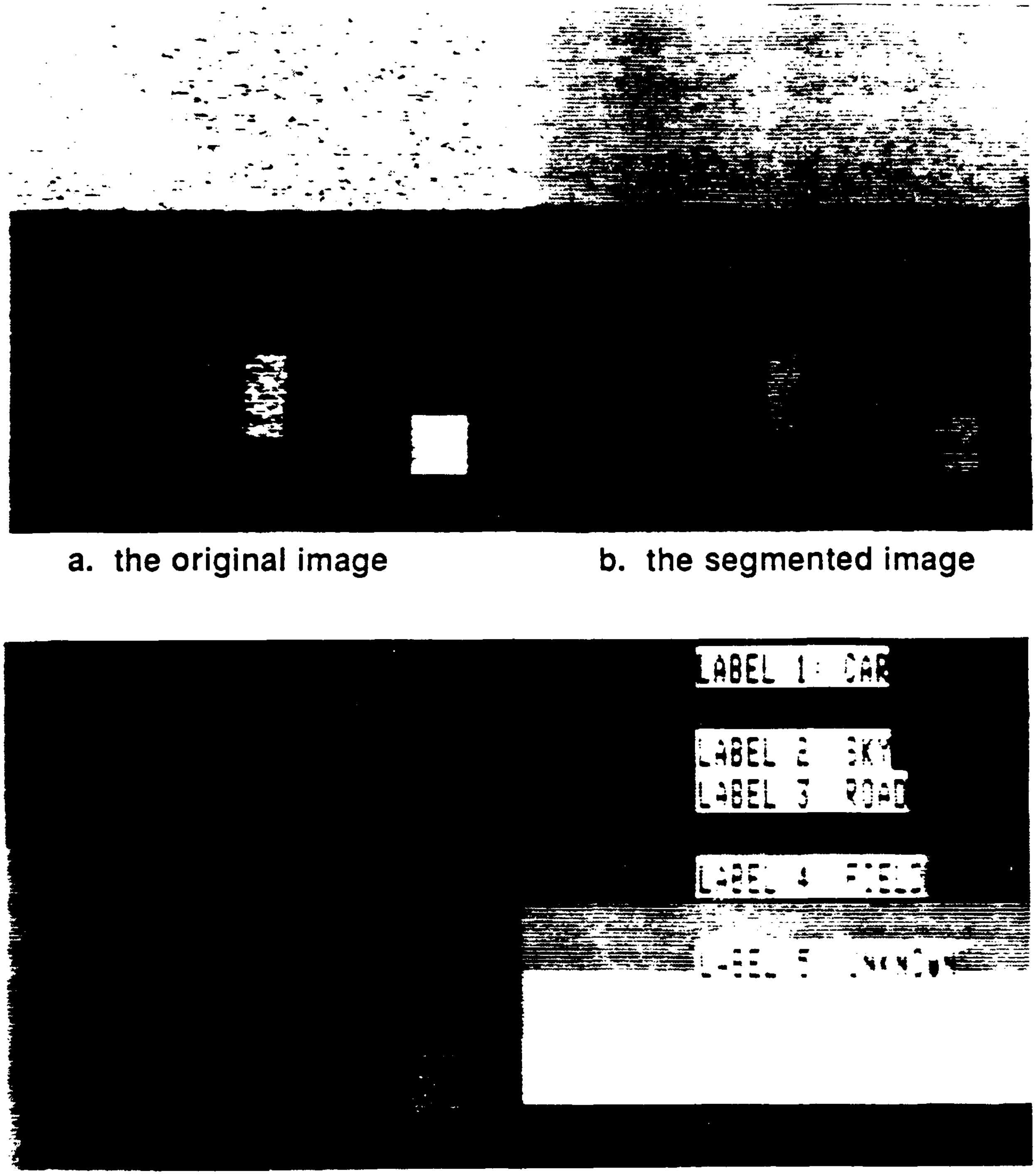
a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

Figure 7.2.6 Interpretation of the Ideal Image with an Unknown Object



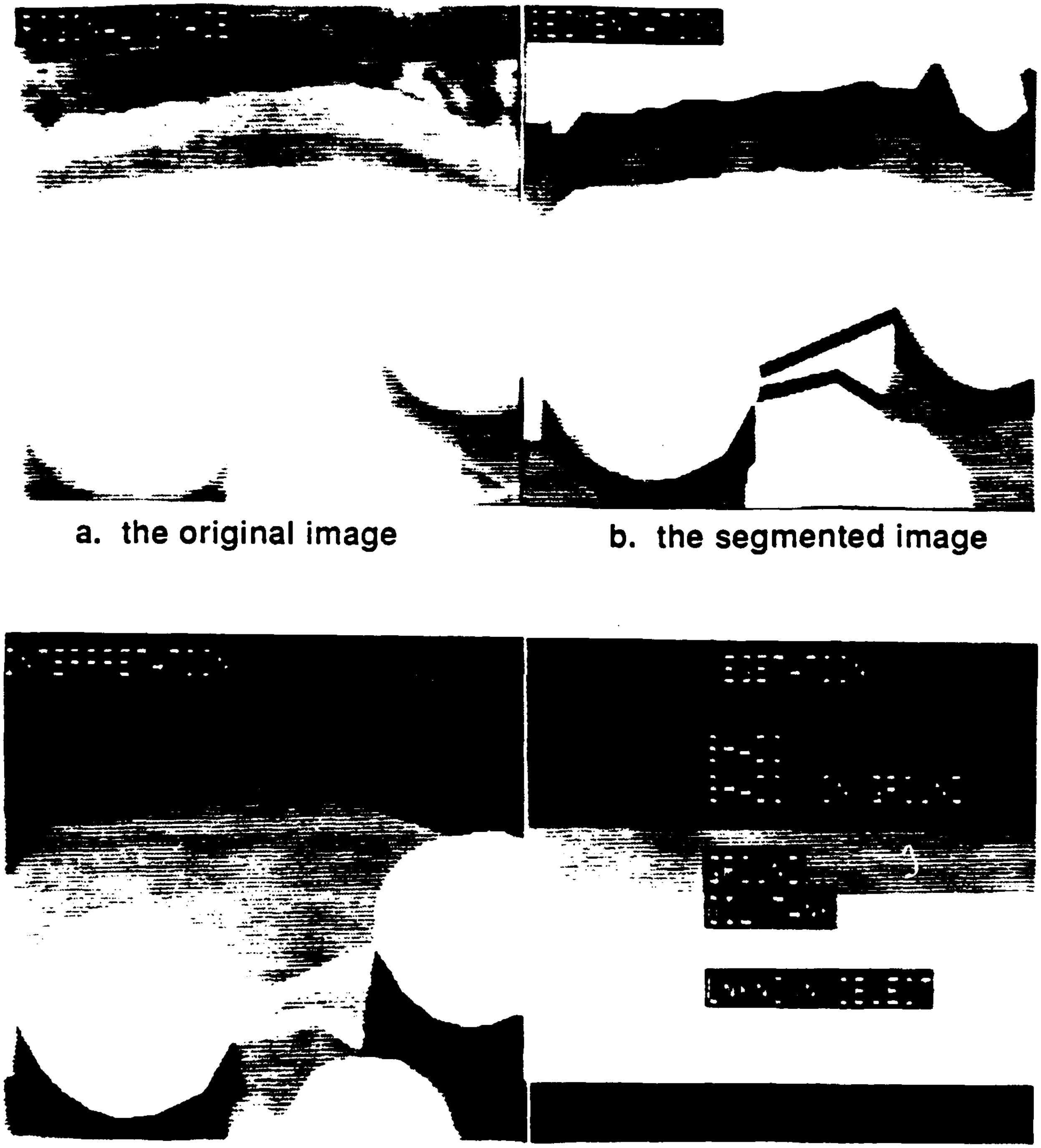
a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

Figure 7.2.7 Interpretation of the 3dB SNR Image
with an Unknown Object



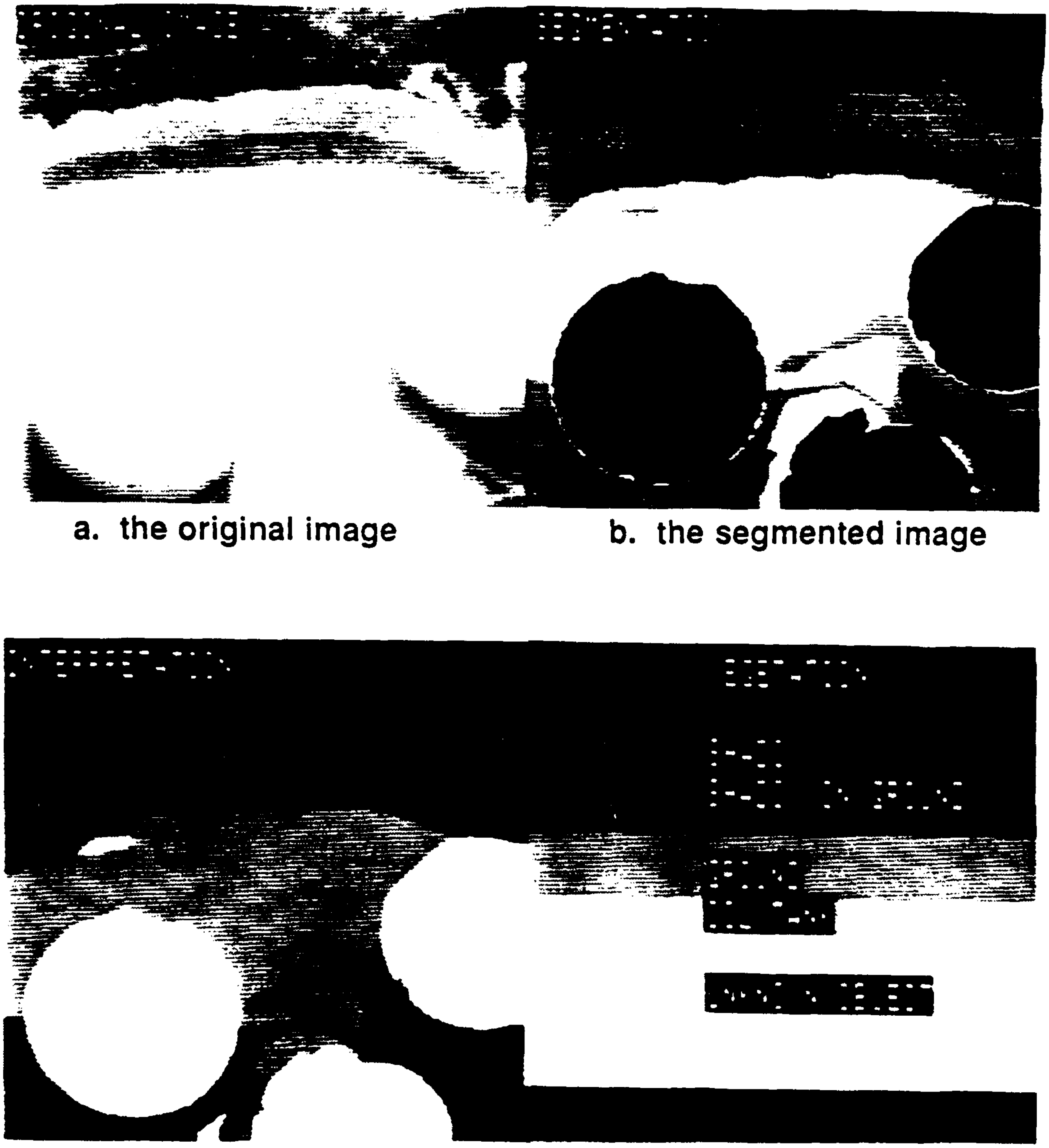
a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

Figure 7.2.8 Interpretation of the Training Image



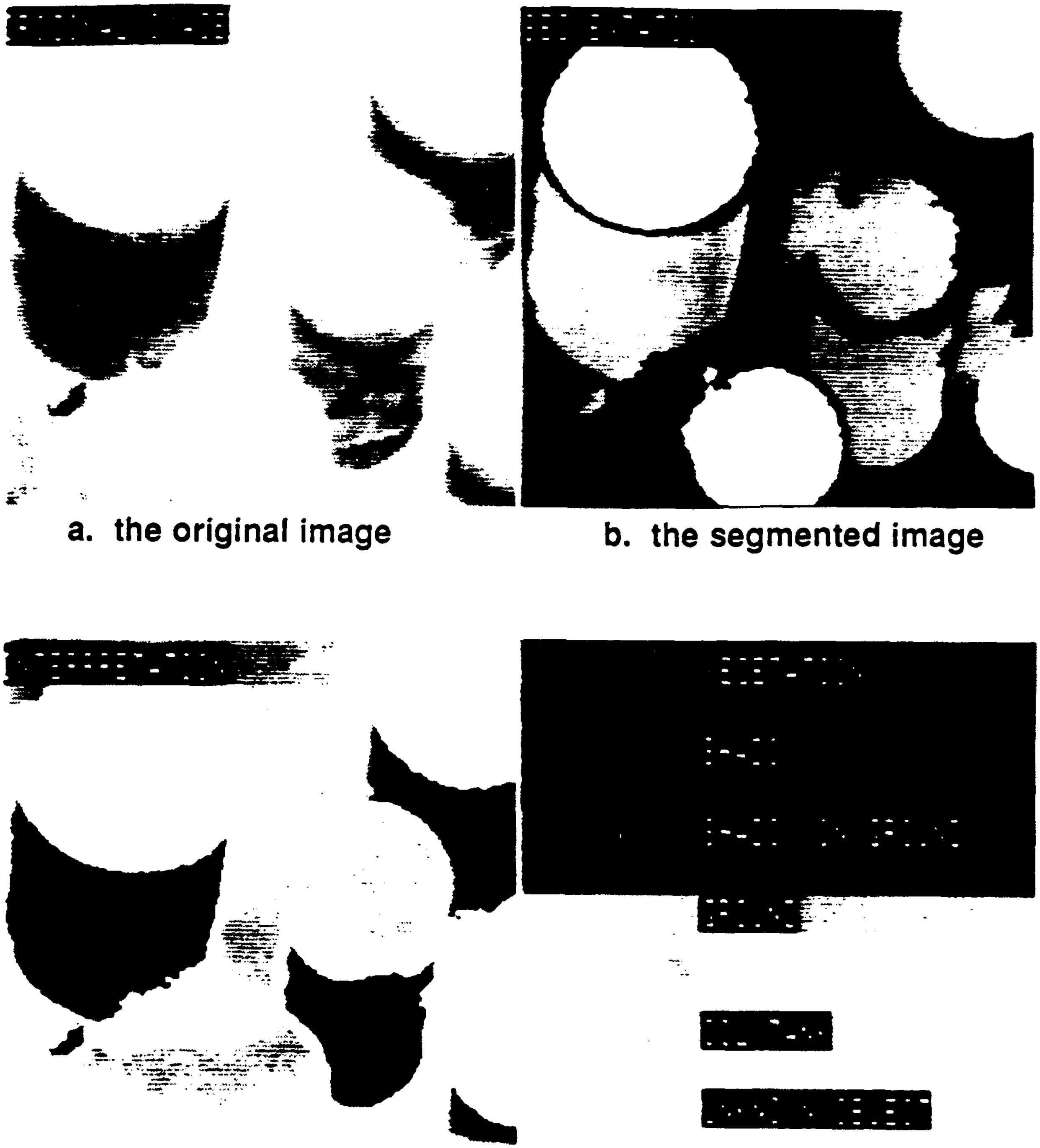
a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

Figure 7.2.9 Interpretation of the Training Image with Computer Segmentation



a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

Figure7.2.10Interpretation of the Test Image with the Compactness Feature

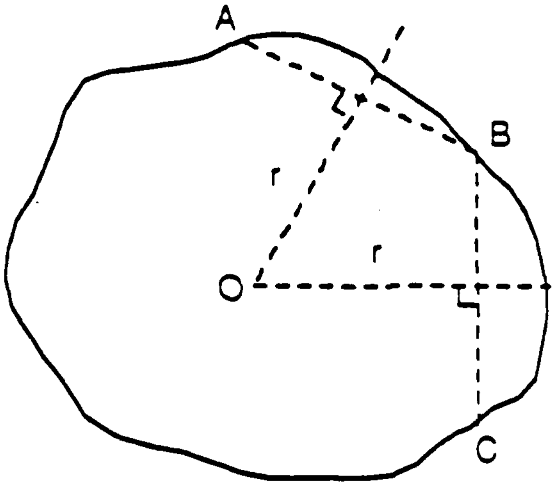
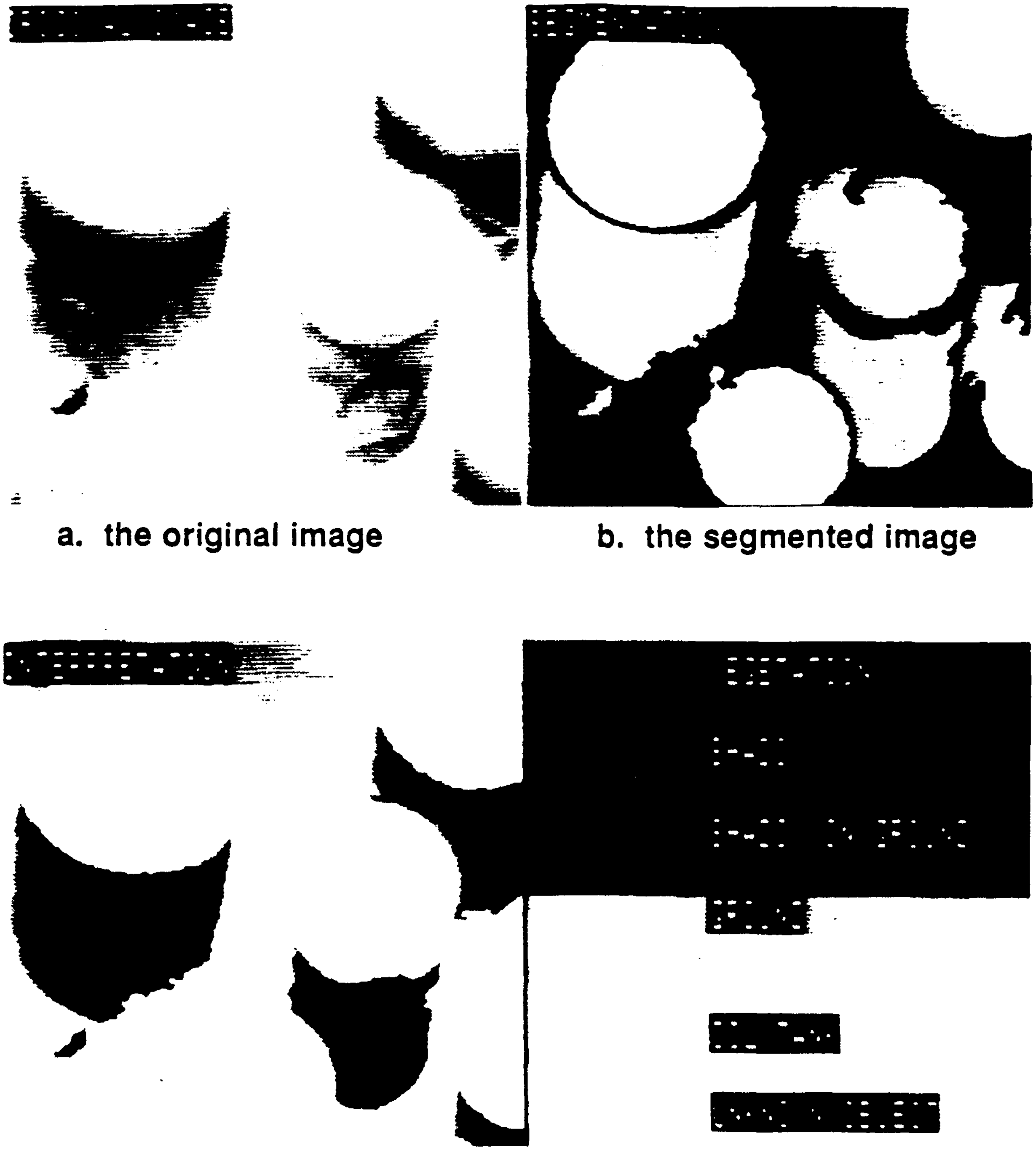


Figure 7.2.11 Illustration of the Partial Compactness Feature.



a. the original image

b. the segmented image

c. the interpreted image

d. gray levels: labels

Figure 7.2.12 Interpretation of the Test Image with the Partial Compactness Feature

7.3 A Model-Fitting Approach to Cluster Validation with Application to Stochastic Model-Based Image Segmentation:

7.3.1 Introduction:

Clustering procedures have found wide application in statistical data analysis and processing. The application of specific interest here is stochastic model-based image segmentation where a clustering algorithm is used to estimate the model parameters for the various image classes in an observed image. In this, and similar applications, it's generally the case that the clustering algorithm requires prior knowledge of the number of clusters or data classes. For many applications, however, the number of clusters is not known a priori and we would like to determine it directly from the data. This is known as the cluster validation problem. For stochastic model-based image segmentation, the solution of this problem directly affects the quality of the segmentation. In this work, we propose a model-fitting approach to the cluster validation problem based upon Akaike's Information Criterion (AIC). The explicit evaluation of the AIC is achieved through an approximate maximum- likelihood (ML) estimation algorithm. We demonstrate the efficacy and robustness of the proposed approach through experimental results for both synthetic mixture data, where the number of clusters is known, and to stochastic model-based image segmentation operating on real-world images, for which the number of clusters is unknown. This approach is shown to correctly identify the known number of clusters in the synthetically generated data and to result in good subjective segmentations in aerial photographs.

7.3.2 Preliminaries:

Clustering procedures are widely used in various applications of pattern classification and statistical data analysis. In a clustering procedure, the observed data or entities are grouped together to form a number of clusters in such a way that the entities within a cluster are more similar to each other than to those in other clusters. The measure of similarity, usually heuristically defined, is called the *cluster criterion*. For example, the Euclidian distance can be used as a similarity measure when the data are finite dimensional vectors.

For the past three decades, many clustering algorithms have been developed by researchers in such diverse fields as biology, statistical data analysis and pattern recognition, using very different cluster criteria [1]. In some previous work [2]-[4] on stochastic model-based image segmentation, clustering algorithms have been used to estimate the model parameter vectors for different image classes directly from the observed image. Since the nature of this work is related to statistical pattern recognition, the clustering algorithm used was selected from those developed within the pattern recognition community. One of the most successful clustering algorithms in this respect is the K-means algorithm [5],[6]. This algorithm is optimum in the sense that it minimizes the variance within each cluster and has been widely used in unsupervised pattern recognition. However, an important problem existing with most clustering algorithms, including the K-means algorithm, is that the *number* of clusters in the data must be specified a priori before using the clustering algorithm.

In some situations this number can be derived from prior knowledge about the data, or sometimes can even be determined from visual inspection of the two- dimensional projection of the data. However, in many applications, such as our previous work on image segmentation, it is desired to estimate this number directly from the observed data since a priori knowledge is generally not available and the data are often vectors of dimension higher than two so that the projection method is not satisfactory. Furthermore, even when the data is two dimensional, visual inspection may not be successful if the data clusters cannot be decided by observation. This problem is of great practical importance for many clustering algorithms and is known as the *cluster validation* problem [7]. For stochastic model-based image segmentation, such as the schemes described in [2]-[4], the solution of this problem directly affects the quality of the resulting segmentation. If the estimated number of clusters, or image classes, is too small, different homogeneous regions in the image will not be well separated. Likewise, if this estimated number is too large, a relatively homogeneous region may be separated into a number of smaller regions. Both of these situations are to be avoided.

There is another class of clustering algorithms, known as hierarchical techniques, that group data in several levels of nested partitions (clusterings). While conceptually they do not require a priori specification of the number of clusters, in practice they require specification of a "cut" level to choose a particular partition which best represents the number of clusters present. The problem here in choosing a "cut" level is then analogous to specifying the number of clusters in the K-means type algorithms and the solution of the later extends easily to the former as discussed in [21].

Most of the previously proposed solutions to the cluster validation problem can be classified into two categories: heuristic approaches and statistical hypothesis testing approaches. In the heuristic approach, the number of clusters is determined by using some ad hoc criteria. For example, for the K-means algorithm a typical approach is to look at the plot of the average of the variances within the clusters under assumptions of different K, the numbers of clusters. The value of K corresponding to the point where the curve begins to saturate can then be taken as the estimated number of classes. Many ad hoc variations of the K-means algorithms have been proposed based on similar ideas. In these algorithms, the number K is increased or decreased according to criteria such as intra-cluster variance and distance between clusters (e.g., the ISODATA algorithm in [6]).

An example of more sophisticated heuristic techniques is the bootstrap scheme for cluster validation proposed recently by Jain and Moreau [21]. In their approach, a number of bootstrap data sets are first generated from the original data set. Then, under the assumption of different number of clusters, the variance of a heuristically defined statistic is computed on bootstrap data sets. The assumed number of clusters that results in the least variance, or believed to provide the most stable clustering, is chosen as the estimate of the number of clusters. Since different clustering procedures may give different solutions using this technique, Jain and Moreau have also defined a heuristic index to determine whether the estimate obtained from a particular clustering procedure is valid. This approach has been demonstrated to provide effective identification of the number of clusters for synthetic data and real data from simple range images. However, as will be discussed later, this method also has some limitations.

Several comprehensive survey studies of various heuristic techniques can be found in [7],[21],[22],[41],[42]. While some practical problems can be solved using the heuristic approach, it does not provide a general solution to the cluster validation problem and, even when applied to specific problems, many techniques have to be fine-tuned through trial-

and-error. This, in part, reflects the difficult nature of the problem. More specifically, as pointed out by Everitt [1] and Jain [7], clusters are generally very difficult to define precisely.

To find generally applicable and mathematically rigorous solutions to cluster validation, many researchers have tried to formulate the problem as a statistical hypothesis testing problem [8],[9],[41]-[43]. For example, hypothesis tests have been proposed to test whether a given cluster should be divided into two. More general likelihood tests have been attempted with the data modeled in terms of finite mixture distributions [9]. However, due to the structure of the mixture distribution, the parameters, which characterize one hypothesis (for example, the null hypothesis) are at the boundary of the parameter space of the other hypothesis. This, in turn, violates the regularity conditions (cf. [9]) which are required for the validity of the asymptotic distribution theory for the generalized likelihood ratio (GLR) test which exists for many simple hypothesis testing situations where each of the hypotheses can be described as a single probability distribution. As a result, no generally applicable GLR test is available at this point to determine the number of clusters directly from observation data.

On the other hand, the problem we face is not unlike the one faced in developing a theory to fit an autoregressive (AR) model to real-world data in which the order of the model has to be decided before the model parameters can be estimated from the data. Having observed that neither heuristic nor hypothesis testing approaches alone would provide a satisfactory solution to determining the order of the model, hence the practical fitting of a model to observation data, Akaike [10] suggested that the problem should be viewed as a *multiple decision* problem. That is, rather than asking which hypothesis is acting (which order is correct), we should ask which model best fits the data. The goodness of fit, as pointed out later by Akaike [11], should be a properly defined entropy function and the best fit should be obtained by maximizing this quantity. Based on this maximum-entropy principle, Akaike proposed a criterion, generally called the AIC (Akaike's Information Criterion), to determine both the order and the parameters of an AR model for observed data. Although there have been some criticisms of the AIC as being inconsistent, Akaike showed that the AIC is robust and optimal in a minimax sense. That is, it is optimal when there is no a priori knowledge about the distribution of the model parameters. In addition, Akaike and others also extended the AIC to several Bayesian variations called the BIC (Bayesian Information Criterion)[13],[14]. This class of criteria can be shown to be AIC's averaged with respect to various a priori distributions for the model parameters. Although the AIC criterion and its variations have achieved substantial success, mostly in AR model fitting, their application is, of course, not limited to AR time series modeling.

There has been little previous work on the application of the AIC to cluster validation. Sclove [17] demonstrated a way to use the AIC to verify image segmentation results. After segmenting a synthetic image under the assumption of two and three classes, the AIC was used to verify that the segmentation with three classes is a better segmentation.

In this work, we have applied the AIC directly to cluster validation and to stochastic model-based image segmentation. Our study has been conducted on both synthetic data and real-world image data. Synthetic data is used to illustrate the efficacy and robustness of the AIC. In particular, we show, through experimental results, that the AIC is effective in correctly identifying the number of classes when the data is generated from well-defined Generalized Gaussian mixtures even for small inter-cluster distances, or large cluster overlap. Furthermore, the AIC can be made quite robust against model mismatch. In the application to image segmentation, we use the AIC to determine the number of distinct image classes modeled by parametric probability models. Our work is different from Sclove's [17] in that we apply the AIC *explicitly* to the cluster validation problem and, in the application to image segmentation, we use the AIC to decide the proper number of classes in an image *before* segmentation.

This paper is organized as follows. In the next section, we will formulate the cluster validation problem as a mixture model-fitting problem and describe how to determine the number of clusters by using the AIC. Then, in Section 7.3.4 and 7.3.5, we discuss how the AIC approach can be applied to cluster validation and image segmentation, respectively. In Section 8/3/7, we demonstrate some experimental results in which the number of clusters, or classes, is determined for synthetic mixture data and images using the AIC criterion. We will also illustrate real-world image segmentation results obtained with the number of classes determined by the AIC. Finally, a summary and conclusions are provided in Section 7.3.7.

7.3.3 The AIC Criterion for Cluster Validation:

Basically, we want to determine the number of clusters by finding the best- fitting random mixture distribution model to the data according to the AIC criterion. Suppose that the sample data can be represented by N independent and identically distributed (*i.i.d.*) random vectors, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$. Furthermore, assume that the data samples are to be described by a mixture distribution. That is, for any $\mathbf{y} \epsilon \mathbf{Y}$,

$$p(\mathbf{y}) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}), \tag{1}$$

where the $p_k(\mathbf{y})$'s are individual mixture-component or *class* probability density functions (pdf's) with π_k, the weights, satisfying

$$\pi_k > 0, \qquad for \quad k = 1, 2, \ldots, K; \tag{2a}$$

and

$$\sum_{k=1}^{K} \pi_k = 1. \tag{2b}$$

Now the problem of cluster validation can be considered as determining K, the number of mixture components, from the observed data where each component corresponds to a distinct cluster. Assuming that the functional form of the component pdf's, $p_k(\cdot)$, are properly selected, we have formulated the problem of cluster validation as that of finding the *best-fitting* mixture model for the data $\mathbf{Y}$. The goodness-of-fit employed here is a properly defined information theoretic criterion, known as the AIC [10], defined by

$$AIC(K) = -2log\{\text{maximum-likelihood of the model } (K)\} + 2K', \tag{3a}$$

where

$$\text{maximum-likelihood of the model } (K) = p(\mathbf{Y}|\hat{\mathbf{a}}_{ML}^{(K)}). \tag{3b}$$

Here, $\hat{\mathbf{a}}_{ML}^{(K)}$ is the maximum-likelihood (ML) estimate of the model parameter vector, $\mathbf{a}^{(K)}$, of the mixture model with K components. Usually, $\mathbf{a}^{(K)}$ contains the component weights in (2) and model parameters for each of the component pdf's, while K' in (3a) is the number of independently adjustable parameters of the K-component mixture model. The AIC approach will select the number of clusters to be K_0, if

$$K_0 = \arg \min_{1 \leq K \leq K_{max}} AIC(K), \tag{4}$$

where K_{max} is a pre-specified upper-limit for K. Using this *minimum AIC principle*, we can compute the $AIC(K)$ for $K = 1, 2, \ldots, K_{max}$, and determine K_0 according to (4). This approach, while shown to be a maximum-entropy principle [10],[11], has a simple heuristic appeal. That is, if two models are about equally likely, the AIC will select the one with smaller number of clusters.

Compared with the previously proposed heuristic techniques, the AIC provides some potential advantages. Take Jain and Moreau's heuristic technique [21] as an example. First of all, although the heuristics used are quite effective in the experimental results provided in [21], little is known concerning how they would work in general; for example, under some specific stochastic modeling assumptions on the data. Secondly, their method is nonparametric while, in many applications, reasonable modeling assumptions can often be made on the data. As a result, parametric methods such as the AIC which make use of more priori information may provide a more precise description of the data. Thirdly, a model-based approach, such as the AIC, provides a more unified and general approach to the cluster validation problem. For example, to achieve effective identification of the number of clusters for different types of data sets, such as hyperspherical mixtures, circular mixtures or enlongated mixtures, Jain and Moreau employed different heuristic statistics as criteria for cluster validation on a case-by-case basis and it does not seem clear how this could be done for other types of mixtures. On the other hand, using a model-based approach, such as the AIC approach, appropriate pdf models can be selected for different types of mixture data and the criterion, in all cases, is based on likelihood functionals. Finally, the computation of the AIC is basically ML estimation which requires only a moderate amount of computation without using any bootstrap data sets which can lead to quite computationally intensive procedures.

7.3.4 The Application of AIC to Cluster Validation:

To apply the AIC approach to a given set of random data, we need to select a mixture distribution model, estimate model parameters and, finally, compute the AIC for different K. We discuss each of these issues in turn.

A.) *The Generalized Gaussian Mixture:*

For simplicity, we further assume the data vectors are m-dimensional and have conditionally independent components given the class. That is, with $\mathbf{y} = \{y^{(1)}, y^{(2)}, \ldots, y^{(m)}\}$, the joint pdf given class k is acting is described by

$$p_k(\mathbf{y}) = \prod_{i=1}^{m} p_k^{(i)}(y^{(i)}); \quad k = 1, 2, \ldots, K. \tag{5}$$

Then the mixture-component, or class, pdf's can be specified through the component pdf's of the data vectors. While the AIC is applicable under quite general mixture models as can be seen from its definition in (3), in this work we have used a particular class of mixture models, known as the *generalized Gaussian* mixture, to demonstrate the procedure of applying the AIC to cluster validation and image segmentation. Let $y^{(i)}, i = 1, 2, \ldots, m$ be a component of data vector $\mathbf{y} \epsilon \mathbf{Y}$. Then a Gaussian component pdf given class k is described by

$$p_k^{(i)}(y^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma_k^{(i)}} exp\left[- \frac{(y^{(i)} - m_k^{(i)})^2}{2\sigma_k^{(i)^2}} \right], \tag{6}$$

where $m_k^{(i)}, \sigma_k^{(i)^2}$ are the mean and variance, respectively. Similarly, a generalized Gaussian componenent pdf given class k is defined for $\alpha > 0$ by

$$p_k^{(i)}(y^{(i)}) = \frac{\alpha \eta_k^{(i)}}{2\Gamma(1/\alpha)} exp\left[- |\eta_k^{(i)}(y^{(i)} - m_k^{(i)})|^{\alpha} \right], \tag{7a}$$

where $m_k^{(i)}$ is the mean, $\Gamma(\cdot)$ is the Gamma function, and $\eta_k^{(i)}$ is a parameter related to the variance, $\sigma_k^{(i)^2}$, by

$$\eta_k^{(i)} = \frac{1}{\sigma_k^{(i)}} \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)} \right]^{1/2}. \tag{7b}$$

When $\alpha = 2$, we have the Gaussian pdf; when $\alpha = 1$, we have the Laplacian or exponential pdf. Plots of a number of typical parametized one- dimensional, zero-mean, generalized

Gaussian pdf's are shown in Fig. 7.3.1 for different values of α. When $\alpha \gg 1$, the distribution tends to a uniform pdf; when $\alpha < 1$, the pdf tends to be more peaked around the mean and also have heavier tails. The latter case results in many"outliers" associated with the corresponding class or cluster. While the Gaussian mixture has often been used to model cluster data, we use the generalized Gaussian model as a convenient model for studing outliers in clusters and evaluating the relative robustness of the AIC.

B.) *Estimation of Model Parameters*:

After the mixture model is selected for the data, the computation in the AIC of (3) mainly involves the ML estimation of the model parameters. The ML estimation approach has been a very successful method in stochastic model parameter estimation for the pdf's which contain only one component. Explicit solution can often be found by solving the appropriate likelihood equation and the ML estimate in many cases is consistent [18]. Even in the case where the true distribution of the data is not the same as the model, the consistency property often still holds under mild regularity conditions [19]. This result is especially important since, when we try to use a model to approximate an unknown probability distribution using ML estimation, it's desirable that the estimates be consistent. Unfortunately, some of these results do not readily extend to mixture distributions [9]. First of all, explicit solution is impossible even for the two-component case. Secondly, the likelihood surface often has singularity points which makes numerical solution difficult. A major reason for this is that the data is *incomplete* in the sense that we do not know a priori to which cluster a data vector belongs. However, a number of approximate ML algorithms for this situation do exist. One of the more popular methods is the so-called EM (expected maximum) algorithm [9], [24]. It has been shown that under mild regularity conditions it does provide local maxima that are consistent [24]. However, a disadvantage of the EM algorithm is its relatively slow convergence [24].

In this work, we use an *approximate* ML estimation scheme using a clustering algorithm. First of all, the K-means clustering algorithm is applied to the data to divide the data into K groups. Then each group is assumed to correspond to the sample data for one and only one mixture component. An ML estimate is then evaluated on each group separately to estimate the parameters for the corresponding mixture component. Finally, the component weights, π_k, $k = 1, 2, \ldots, K$, can be estimated as the ratio of the number of samples in a group to the total number of samples. This approximation transforms the problem of ML estimation of a mixture to that of ML estimation of several individual pdf's. In the next section, we demonstrate, through experimental results, that it provides reasonably good estimates. This scheme also converges fast since the underlying clustering algorithm is known to possess fast convergence properties.

There are two points that need to be noted in using the K-means algorithm for mixture estimation. First, it has been pointed out by Titterington [23], among others, that *theoretically*, the K-means algorithm results in asymptotically biased estimates. However, we found that this did not seem to effect the performance of the AIC approach. As a matter of fact, the K-means algorithm provides reasonably good estimates and the AIC computed using this algorithm is quite effective in identifying the number of classes correctly in a variety of experiments to be described later. In addition, compared to the EM algorithm, the K- means algorithm is computationally more efficient. Hence, in the results of this paper, we have used the K-means algorithm exclusively, although the use of the EM algorithm for the AIC is currently under investigation. The second point is how to chose the initial cluster centers, or *seeds*, when using the K-means algorithm since the result of the clustering, being locally optimum, often depends on the choice of seeds. For example, Jain and Moreau [21] suggest using several different sets of randomly selected seeds when the K-means algorithm is used for a given specification of the number of clusters. While this improves the data clustering, or classification, the amount of computation needed is increased drastically. In our experiments, we have found that the results of parameter estimation and the subsequent cluster identification using the AIC is relatively insensitive to whether the K- means algorithm is used with one set or more than one set of random seeds. Hence, in all the experiments, we use only one set of seeds selected from the data randomly when using the K-means algorithm for a given K. In summary, the K-means algorithm is used as a computationally efficient and reasonably accurate estimation procedure. While it works well for our purposes, we do not claim it is the best procedure.

In the above clustering-estimation procedure, once the data are classifed, the model parameter estimation problem reduces to obtaining ML estimates for the parameters of each of the mixture components. For example, in the Gaussian mixture case, it is well-known [18] that the ML estimate of the mean is

$$\hat{m}_k^{(i)} = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{k_j}^{(i)}, \tag{8a}$$

where we suppress the subscript "ML" for convenience, while similarly

$$\hat{\sigma}_k^{(i)^2} = \frac{1}{N_k} \sum_{j=1}^{N_k} (y_{k_j}^{(i)} - \hat{m}_k^{(i)})^2 \tag{8b}$$

represents the ML estimate of the variance. Here, N_k is the number of samples assigned to class k, while the $y_{k_j}^{(i)}$'s represent the i'th component of the sample vectors assigned to class k. The ML estimate of the mean of the generalized Gaussian is the same as (8a), while the ML estimate of the variance can be conveniently related to the ML estimate of η through (7b). More specifically, in Appendix A, we show that

$$\hat{\sigma}_k^{(i)^2} = \left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right] \left(\frac{\alpha}{N_k} \sum_{j=1}^{N_k} |y_{k_j}^{(i)} - \hat{m}_k^{(i)}|^\alpha\right)^{2/\alpha}, \tag{9}$$

which clearly reduces to (8b) for $\alpha = 2$.

C.) *Computation of the AIC*:

There are two applicable expressions for the likelihood functional when using the AIC criterion. If we consider the data vectors to be *incomplete*, that is, the class status of the samples is unknown, we will have the standard likelihood expression for the mixture which, from (1), becomes

$$p_K(\mathbf{Y}|\mathbf{a}) = \prod_{i=1}^{N} \sum_{k=1}^{K} \pi_k p_k(\mathbf{y}_i). \tag{10}$$

On the other hand, if we first classify the data by applying the K-means algorithm, we in effect assign data vectors to hypotheses classes. In this case a data vector assigned to class k can be considered coming from a particular class and has a probability π_k of occurring. The corresponding expression for the likelihood functional for *correctly* classified samples, or *complete* data, then becomes

$$p_K(\mathbf{Y}|\mathbf{a}) = \prod_{k=1}^{K} \pi_k^{N_k} \prod_{j=1}^{N_k} p_k(\mathbf{y}_{k_j}), \tag{11}$$

where $N_k \leq N$ is the number of samples in the k^{th} cluster and $\mathbf{y}_{k_j}, j = 1, 2, ..., N_k$ are data vectors associated with this cluster [9], [23]. Since we have used the K-means algorithm for approximate ML estimation, each sample vector is assigned to a unique class. In what follows, we will make use of the second likelihood functional as expressed by (11). The ML estimate, $\hat{\mathbf{a}}_{ML}^{(K)}$, to be used in (3) in computing $AIC(K)$ is then formed from the resulting K class-conditional parameter estimates together with the estimated weights as described above.

Now, the application of the above model-fitting approach to random data to determine the number of clusters is straightforward, as described by the following steps:

1.) Start with $K = 1$.

2.) For each $K = 1, 2, \ldots, K_{max}$, compute $AIC(K)$ using the approximate ML estimation procedure.

3.) Announce that there are K_0 clusters if $AIC(K_0)$ is the minimum among all $AIC(K)$'s obtained in step 2.).

7.3.5 Application of the AIC to Image Segmentation:

The model fitting approach to cluster validation can also be applied to stochastic model-based image segmentation. In a stochastic model-based approach, an image is segmented into regions of different statistical properties, or *classes*, that are modeled by parametric random field models. In [27], we have considered simple independent Gaussian random field models for the image classes. More specifically, let the image be a two-dimensional (2-D) array on the lattice L, denoted by $\mathbf{x}$, and described by

$$\mathbf{x} = \{x(i,j), (i,j) \epsilon L\}, \quad L = \{(m,n), 1 \leq m, n \leq N\}. \tag{12}$$

Then an independent Gaussian random field can be represented by

$$x(i,j) = f + w(i,j); \quad (i,j) \epsilon L, \tag{13}$$

where f is a constant mean, $w(i,j)$ is a 2-D zero-mean Gaussian white noise process with variance σ_w^2. In this paper, we will also consider two other image models; the AR model and the Markov random field (MRF) model [28]-[30]. In particular, we consider a simple AR model, known as the first quarter-plane filter (FQPF) configuration, represented by

$$x(i,j) = ax(i-1,j-1) + bx(i-1,j) + cx(i,j-1) + w(i,j); \quad (i,j)\epsilon L, \qquad (14)$$

where a, b, c are constant model coefficients, representing, respectively, the diagonal, horizontal and vertical correlation between pixels, and $w(i,j)$ is again a 2-D zero-mean white Gaussian noise process with variance σ_w^2. The MRF model considered in this paper is also very simple, known as the binary isotropic auto (BIA) model [28]. It is represented by the joint pdf

$$p(\mathbf{x}) = Z^{-1} exp[-U(\mathbf{x})], \qquad (15a)$$

where $U(\cdot)$ is the *Gibbs energy functional*, defined as

$$U(\mathbf{x}) = a\sum_{i,j} x^2(i,j) + b\sum_{i,j} x(i,j)[x(i-1,j) + x(i,j-1)], \qquad (15b)$$

where the $x(i,j)$ terms are binary and Z is a normalization factor to make (15a) a valid pdf. For this model, a and b are the constant model parameters that control the average level, and the horizontal and vertical correlation of the pixels, respectively. Since our purpose here is to demonstrate the applicability of the AIC to different image models, the models selected are rather simple. However, the method developed here is applicable to more complex image models as long as they are parametric.

Under appropriate stochastic modeling assumptions for the different image classes, the image segmentation problem can be formulated as a statistical decision problem, where each pixel is assigned to one of a finite number of classes. Typical techniques of stochastic model-based segmentation include ML and MAP segmentation procedures [2]-[4], [31]-[37]. Before applying such a technique to an image, however, the model parameters for each class have to be estimated. In an *unsupervised* approach, which is the concern here, these parameters are estimated directly from the image. However, to estimate the model parameters, we first need to know *how many* classes there are in the image. The AIC-based model-fitting approach described previously can be applied directly in determining the number of classes in an image. Indeed, in this case, the algorithm outlined previously for computing the AIC's can be applied once the corresponding random data vectors are specified and obtained. Rather than using all the pixels in the image (there are just too many), the sample data vectors are chosen to be the *estimated* model parameter vectors obtained from different spatial positions of a sliding estimation window on the image, as shown in Fig. 7.3.2. Here, M_1, N_1 determine the size of the rectangular window, while M_2, N_2 are the vertical and horizontal displacements, respectively, between two adjacent spatial window positions. Hence, we perform cluster validation in the model parameter vector space instead of the pixel space. The number of clusters, or image classes, would then be the number of distinct clusters of different model parameter vectors. This scheme was proposed in our previous work [27], under the assumption that the image classes are modeled as independent Gaussian random fields; in which case the model parameter vectors are two-dimensional, consisting of only the mean and variance computed within a sliding estimation window on the image. In this paper, we will continue to use this scheme. However, we will also consider more complex models for the image classes such as the AR and the MRF models in which case the model vectors need no longer be two-dimensional. For example, the FQPF AR model has four-dimensional model parameter vector (a, b, c, σ_w); the BIA MRF model, on the other hand, has two-dimensional model parameter vector (a, b). Again, we emphasize that this scheme does not have any restrictions on the models for the image classes, as long as they are parametric. Once the model parameter vectors are so obtained, the minimum AIC principle described above readily applies.

7.3.6 Experimental Results and Discussion:

In this section, we will present and discuss some experimental results on the model-fitting approach utilizing the AIC applied to synthetic mixture data and various stochastic and real-world image data. For synthetic data, we concentrate on the performance of the AIC as a cluster validation criterion, e.g., its efficacy under different data distributions, its performance under various degrees of overlap of the clusters and its robustness under model mismatch (in the sense when the model structure assumed in the AIC is different from the actual data distribution). More specifically, we divide the study on synthetic data into three parts. In the first part, we apply the AIC to identify the number of clusters in different Gaussian mixtures where the data clusters are relatively well separated. In the second part, we study the performance of the AIC for generalized Gaussian mixtures under varing degrees of cluster overlap. In the third part, we study the relative performance when the assumption of the model structure in the AIC is different from the actual data distributions, again using the generalized Gaussian mixtures for illustration.

The study of the AIC on image data concerns the identification of the number of image classes in synthetic or real images, where the image classes are modeled as random fields such as Gaussian, AR and MRF's. In this part, we have used both synthetic images and real-world images. The synthetic images are realizations of mixtures of the previously mentioned random field models and the number of clusters is specified when they are generated. In the experiments, these synthetic images are used to demonstrate the efficacy of the AIC in identifying the number of classes correctly when the number of classes is well defined. For the real-world images we have used aerial photographs. In these images, usually the true number of classes is not clear or well-defined as in the case of synthetic images and the AIC is used to provide an *objective* way of determining the number of classes as opposed to *subjective* assessment. Notice that in this case, the various classes are statistical models which may or may not correspond to distinct real-world objects. The AIC criterion is used only to help segment the image into regions that are *statistically different*, while object recognition and global image interpretation are usually performed subsequently using the segmented image together with domain knowledge and measurements made on the segmented regions [37].

A.) *Synthetic Data: Gaussian Mixtures*:

In this experiment, three two-dimensional ($m = 2$) Gaussian mixture data sets with two, three and four components, or clusters, are generated as shown in Fig. 7.3.3. We choose the data to be two-dimensional since it's then easy to display on a plane. There are two objectives of this experiment: first, to see if the approximate ML estimates provide a reasonable estimate of the true model parameters and, secondly, to see whether the

AIC provides correct estimates of the number of clusters, even in this idealized case. The results of the parameter estimates for all the test data sets under the correct assumptions on the number of clusters are shown in Table 7.3.1. It can be observed that when the assumption of the number of classes acting corresponds to the true but unknown value, the parameter estimates are quite accurate. This is clearly the case here for the data clusters which are reasonably well- separated, and also true for other experiments on synthetic data, described below, where the clusters are overlaped. The K-means algorithm, which may not be unbiased theoretically, does provide a reasonably effective and fast procedure for mixture estimation and likelihood computation. This indicates that the approximate ML estimation scheme using clustering is quite effective. In Table 7.3.2, we have shown the AIC's computed for all the test data under the assumptions of different numbers of clusters, with $K_{max} = 8$. We find that the AIC does make correct decisions each time. This indicates that when the data is indeed a Gaussian mixture and the clusters are well-separated, the method proposed here tends to estimate the number of clusters correctly. Additional examples are given for a much larger variety of Gaussian mixtures in [19] with similar results.

B.) *The Performance of the AIC versus the Distance between Clusters*:

Here, we are interested in the performance of the AIC on a variety of data sets as a function of the distance between clusters; in particular, when the distance is successively reduced. For this purpose, synthetic two-dimensional random mixture data are generated from a number of generalized Gaussian mixture models. For simplicity, in the generalized Gaussian mixture model for each data set, there are three mixture-components with equal component probabilities. That is, for each data set, $K_{truth} = 3$, and $\pi_1 = \pi_2 = \pi_3 = 1/3$. In addition, we set the variances of each mixture-component to be 1.0 for all the data sets; that is, ${\sigma_k^{(i)}}^2 = 1.0$, $k = 1, 2, 3$; $i = 1, 2$, for all data sets. In Fig.'s 7.3.4-7.3.7, we have shown the 16 data sets generated for this experiment. In each figure, there are four data sets generated with the same α, but with different distances between the three clusters. Hence, the data sets in these figures are different in two ways. From one figure to another, they differ in α, the *characteristic exponent* of the underlying generalized Gaussian model; within each figure, they differ in d, the normalized distance between clusters. The normalized distance, d, for a data set is defined as

$$d = \frac{\max_{i,j} D(\mathbf{c}_i, \mathbf{c}_j)}{\sqrt{\delta_i \delta_j}}, \tag{16a}$$

where $\mathbf{c}_k$ is the mean, or cluster center, of the k'th cluster given by

$$\mathbf{c}_k = \mathbf{m}_k = (m_k^{(1)}, m_k^{(2)}, \ldots, m_k^{(m)}); \quad k = 1, 2, 3, \tag{16b}$$

and

$$D(\mathbf{c}_i, \mathbf{c}_j) = ||\mathbf{c}_i - \mathbf{c}_j|| \tag{16c}$$

is the Euclidian distance between two cluster centers. ${\delta_k}^2$ is the average intra-cluster variance, defined as

$${\delta_k}^2 = \frac{1}{m}\left[\sum_{i=1}^{m} {\sigma_k^{(i)}}^2\right]; \quad k = 1, 2, 3. \tag{16d}$$

It is used, in general, to take into account the spread of the clusters. For the two-dimensional data sets considered here, $m = 2$ and $\delta_k = 1$ for $k = 1, 2, 3$. Finally, in each data set, the cluster centers are chosen to be on a circle centered at (15,15) with radius r and 120° apart. Hence, in all the data sets, $d = \sqrt{3}r$ and is proportional to r.

The AIC's computed under the *assumption* of a generalized Gaussian mixture model with $\alpha_0 = 3.0, 2.0, 1.0, 0.5$, respectively, are computed for each data set in Fig.'s 7.3.4-7.3.7, with $K_{max} = 5$. The results are shown in Tables 7.3.3-7.3.6, respectively. Examining the resulting AIC's, we can see that the AIC with different assumptions on α, denoted by different α_0's, correctly identifies the number of clusters for most of the data sets in Fig.'s 7.3.4-7.3.7 with cluster distances of $r = 5.0, 2.5, 2.0$; it breaks down only for data sets with cluster distance of $r = 1.5$, when the clusters are so close that they are hard to distinguish visually. Interestingly, however, in the case of $r = 1.5$, when the AIC provides erroneous decisions, it generally suggests that there is only one cluster. An exception is the case illustrated in Table 7.3.6 where the data is generated from a generalized Gaussian mixture with $\alpha = 0.5$, as in Fig. 7.3.7. In this case, the data clusters contain many "outliers";

that is, data points that belong to a cluster but are relatively far from the cluster center. The AIC computed based on assumptions of $\alpha_0 = 3.0, 2.0$, and sometimes 1.0, makes erroneous decisions even for the data set with $r \geq 2.0$ where the clusters are visually quite distinguishable. On the other hand, the AIC's computed under the assumption of $\alpha_0 = 0.5$ still provide correct decisions, even for $\alpha = 0.5$ and $r = 1.5$. In the latter case, the clusters are still visually distinguishable, since $\alpha = 0.5$ represents a pdf very peaked around the mean and with heavy tails.

From these results, we can observe that when the data clusters are from the generalized Gaussian class, and they do not contain many outliers (e.g., $\alpha \geq 2$), the AIC not only makes correct decisions when the clusters are far apart, it remains effective when the clusters are relatively close, and only breaks down when the clusters tend to merge as one, in which case it reasonably indicates that there is one cluster. When the data does contain many outliers (e.g., $\alpha < 1$), the AIC computed based on a heavy-tailed generalized Gaussian component pdf assumptions (e.g., $\alpha_0 = 0.5$) still provides correct results. This brings up the next subject of discussion.

C.) *Relative Robustness*:

In this part, we first study the robustness of the AIC under different generalized Gaussian modeling assumptions; namely, the performance of the AIC when the assumed value, α_0, of the characteristic exponent differs from the true value, α. Some indication of this behavior can already be seen from Tables 7.3.3-7.3.6. For example, taking the Gaussian assumption as an illustration, we can observe that this choice (i.e., $\alpha_0 = 2$) is quite robust as long as there are not too many outliers in the data clusters, i.e., provided $\alpha > 1$. It breaks down quickly when there are too many outliers in the data clusters, i.e., $\alpha < 1$. Similar observations can be made for the case of $\alpha_0 = 3$ and $\alpha_0 = 1$.

An assumption of $\alpha_0 < 1$, on the other hand, is quite robust. For example, the assumption of $\alpha_0 = 0.5$ provides very robust performance and correct identification for all the data sets with $r = 5.0, 2.5, 2.0$. Furthermore, with $r = 1.5$ it either provides correct identification or suggests that there is only one cluster, which is quite reasonable.

To compare the relative robustness of the Gaussian assumption with that of a heavy-tailed pdf (e.g., $\alpha_0 < 1$), it is also instructive to look at the way variances of the mixture component pdf's are estimated under different assumptions on α_0. Notice that, from expression (8b), when there are many outliers, the estimate of the variance under the Gaussian assumption will be degraded due to the influence of the outliers as a result of the squaring operation on the data samples. On the other hand, from (9), when we adopt an assumed generalized Gaussian model with $\alpha_0 < 1$, the effect of the outliers in the estimate of variance would be less compared to the Gaussian case ($\alpha_0 = 2$). In Appendix B, we show that the *log* maximum-likelihood functional in the AIC of (3) for a given data set depends only on the *variances* of the mixture components, and *the number of data points* assigned to each of the classes during the ML estimation process. In particular, from expression (B8) of Appendix B, large intra-cluster variances tend to reduce the log likelihood, or increase the AIC for a given K, the number of classes. Therefore, when there are a considerable number of outliers in the true data clusters, they will affect the estimate of the intra-cluster variances and hence the computed AIC and the corresponding estimate of the number of classes. Furthermore, the outliers tend to affect $AIC(K)$ more for smaller K's. Consider the case, for example, where there are actually two clusters in the data, relatively far apart but each containing a considerable number of outliers. When we compute $AIC(K)$ for $K = 2$ by the approximate ML algorithm described previously, the K-means algorithm is used first to separate the data into two clusters. Suppose that the separation is reasonably close to what the two clusters should be. Then, in each of the clusters, the outliers, being far from the cluster center, would tend to increase the estimate of the variance of the clusters, hence increasing the AIC. On the other hand, when we compute $AIC(K)$ for $K > 2$, the outliers in question will be closer to some cluster centers than in the case of $K = 2$, since the new clusters are now formed from the original two data clusters in the case of $K = 2$. As an extreme case, when K is equal to the number of data points, all the data points will be cluster centers and the intra-cluster variance will be zero. To summarize, if we compute the AIC under the Gaussian assumption, the outliers will tend to affect the efficacy of the AIC, and this affect is more severe when the actual number of data clusters is small. For other generalized Gaussian assumptions, the outliers also affect the computed AIC's, and the effects on assumptions for $\alpha_0 > 1$ is greater than when $\alpha_0 < 1$. Next, in order to provide more insight into the robustness problem, we look at it in the light of the theory of robust statistics.

Up to now, we have used the word robust in a loose sense. In particular, we have said that a specific modeling assumption, for example, a generalized Gaussian mixture assumption (e.g., $\alpha_0 = 0.5$), is robust if the AIC computed based on it performs well when the actual data is from a different distribution; for example, a Gaussian mixture. However, the word "robustness" has a more strict meaning in the theory of robust statistics [38],[39]. More specifically, an estimate of a parameter, such as the mean or variance, computed under a given modeling assumption, is called robust if it is relatively insensitive to small *deviations* of the actual data distribution from the assumed model. In our previous examples using the synthetic data in Fig.'s 7.3.4-7.3.7, the AIC's computed under the generalized Gaussian modeling assumption of $\alpha_0 = 0.5$ are quite insensitive to the actual data distribution, for example, for $\alpha = 3, 2, 1$. Here, the *deviation* of the model is characterized by deviations in α, the *characteristic exponent* of the distribution. However, in the theory of robust statistics, more general deviations from the assumed model are considered. For example, let the assumed model for the observed random variable, X, be a pdf denoted by $f_0(x)$. A small deviation from the assumed model can be described as the mixture pdf

$$f(x) = (1 - \epsilon) f_0(x) + \epsilon g(x), \tag{17}$$

where $g(x)$ could be any valid pdf, while ϵ is a small positive number that describes the fraction of gross error in observed data.

The theory of robust statistics concerns finding *optimal* robust estimators. Two important criteria for optimality are the *minimax* principle and the *gross-error-sensitivity* criterion. The minimax principle provides the most efficient estimate of the given parameter for the worst deviation from the assumed model [38]. The gross-error-sensitivity criterion provides the most efficient estimate under a given upper-bound of gross-error-sensitivity. It has been shown that these two approaches provide the same results for a number of important problems, such as the robust estimation of mean and variance [39]. In this work, we use some of the methods of robust statistics, in particular, methods for robust variance estimation, to obtain more insight into the problem of robustness in computing the AIC under different modeling assumptions. More specifically, we seek alternative methods for computing the AIC such that the efficacy will be preserved when the data classes contains outliers and the actual data distribution deviates from the modeling assumptions. Comprehensive treatment of the theory of robust statistics and its various applications can be found in [38]-[40].

In this work, we take the following approach for robust estimation of the AIC. We assume the data pdf deviates slightly from a Gaussian mixture due to outliers in each of the actual data clusters. More specifically, we assume that the pdf of each of the mixture components deviates from an assumed *nominal* Gaussian pdf. This can also be described by expression (17), where $f_0(\cdot)$ is the Gaussian mixture component pdf and $g(\cdot)$ could be any valid pdf. Under the Gaussian assumption, as pointed out previously, the log likelihood of the $AIC(K)$, for a given K, depends only on the intra- cluster variances and the number of data points in each of the K clusters resulting from the K-means clustering procedure. Hence, we should be able to achieve robust estimates for the AIC's through using robust estimates of the intra-cluster variances. In this approach, when the AIC is computed, we still use the expression based on a Gaussian assumption but we replace the variance estimates of (8b) by robust estimates. Since we assume the components of the data vectors are independent, we can look at the robust variance estimation for individual components of the observed random data vectors of a given class.

The approach we take is based on the so-called *influence functions* as treated in [39]. Consider the problem of estimating a parameter (e.g., mean or variance) of an underlying distribution from a number of $i.i.d$ observations; for example, the ML estimation problem in (8)-(9) of Section 7.3.4. An influence function is defined for a random observation, a parameter, and a pdf. The pdf is usually the *ideal*, or nominal, model for the random variable; for example, $f_0(\cdot)$ in (17) from which the *real* data pdf deviates. The influence function describes the asymptotic effect of a contaminated observation, or outlier, on the estimate of the parameter (for example, the variance) for an assumed ideal pdf model. More specifically, let the contaminated observation be denoted by x, the ideal pdf be denoted by f_0 and the estimate of the statistic from a finite number of samples by T_n, where n is the number of samples. Furthermore, assume the estimates under consideration are Fisher consistent under the ideal model, i.e., when f_0 is acting, $\lim_{n\to\infty} T_n = T$ in probability, where $T = T(f_0)$ can be considered a functional of the underlying distribution. Usually, this functional can be expressed as an expectation (or a function of it) with respect to this underlying distribution. For example, in the case of the ML estimate of variance in (8b), we have

$$\begin{aligned} T = T(p_k^{(i)}) &= \int_{-\infty}^{+\infty} y^2 p_k^{(i)}(y)dy \\ &= E_{p_k^{(i)}}\{Y^2\}, \end{aligned} \qquad (18)$$

or, for simplicity, $T = \sigma_k^{(i)^2}$. Then, in general, the influence function can be defined in terms of x, T, and f_0 as [39]

$$INF(x, T, f_0) = \frac{\partial}{\partial t}[T((1-t)f_0 + t\delta_x))] \,|_{t=0}, \qquad (19)$$

with T considered an appropriately defined functional of the indicated pdf, while δ_x is the delta-function pdf with the unit probability mass concentrated at x. In addition, T must also be assumed Fisher consistent under the mixture pdf $(1-t)f_0 + t\delta_x$ for $0 \le t \le 1$.

The influence function provides insights into the relative robustness of an estimate of a parameter. For example, the influence function of the ML estimate of variance of (8b) under a zero-mean unit-variance Gaussian assumption for f_0 can be shown to be [39]

$$INF(x, T, f_0) = x^2 - 1. \qquad (20)$$

From (20), we can see that when a contaminated observation is far from the mean (which is zero here), it has a large effect on the resulting estimate of variance, T, due to the squaring operation in the ML estimate of variance. This agrees with the intuitive observation made in previous sections.

The component variance is, in general, a measure of the intra-cluster spread, or dispersion. Often it is possible to develop alternative measures of intra-cluster spread which are less sensitive to outliers and can be utilized in place of the ML estimate of variance in computing the AIC. For example, suppose we measure the intra-cluster spread in terms of the *mean absolute deviation*

$$\hat{\sigma}_k^{(i)} = \frac{1}{N_k} \sum_{j=1}^{N_k} |y_{k_j}^{(i)} - \hat{m}_k^{(i)}|, \tag{21}$$

and then square this to obtain the statistic $\hat{\sigma}_k^{(i)^2}$. In general, this is a biased estimate of the true component variance, $\sigma_k^{(i)^2}$. Nevertheless, it is often the case that asymptotically this statistic converges to a quantity proportional to the true variance with the constant of proportionality independent of the data. This is the case, for example, with the generalized Gaussian component distribution. It follows then from (B8) that asymptotically the log-likelihood functional computed using the statistic $\hat{\sigma}_k^{(i)^2}$ given by (21) is *equivalent* to that using the ML estimate of variance. This follows by arguments used in Appendix B in demonstrating the equivalence of (B11) and (B8) by eliminating data independent terms. This then provides the rationale for using alternative statistics, such as given by (21), in place of ML estimates. Furthermore, this statistic has an influence function

$$INF(x, T, f_0) = 2\sqrt{\frac{2}{\pi}} \left(|x| - \sqrt{\frac{2}{\pi}} \right), \tag{22}$$

as is shown in Appendix C. Here, $T = E_{f_0}^2\{|Y_k^{(i)}|\}$, where the subscript indicates the expectation is performed with respect to the pdf f_0. Compared to the ML estimate, the effect of the outlier is smaller. Indeed, it has been shown [38] that (21) is a more robust estimate of variance when the real pdf deviates from the ideal Gaussian assumption although it is less efficient under the ideal Gaussian assumption.

Now, this statistic can be extended to a simple *heuristic* method for estimation of the intra-cluster spread to reduce the effects of the outliers. This estimator is given by

$$\hat{\sigma}_{k,h}^{(i)^2} = \left[\frac{1}{N_k} \sum_{j=1}^{N_k} |y_{k_j}^{(i)} - \hat{m}_k^{(i)}|^{\beta} \right]^2, \tag{23}$$

where $0 < \beta \leq 1$, and the subscript h indicates "heuristic". The corresponding influence function is

$$INF_h(x, T, f_0) = 2A(\beta)\big(|x|^{\beta} - A(\beta)\big), \tag{24a}$$

where $A(\beta)$ is a constant for specified β, given by

$$A(\beta) = E_{f_0}\{|Y|_k^{(i)^\beta}\}. \tag{24b}$$

Here, $T = A(\beta)^2$. In the next part, we will show that this heuristic scheme provides some interesting results for identifying the number of classes in images. Finally, the influence function for the *optimal* robust estimator (in the sense of both minimax and gross-error criterion) in this case is

$$\begin{aligned} INF(x, T, f_0) &= x^2 - 1; \quad & x^2 - 1 \leq b; \\ &= b; \quad & x^2 - 1 > b, \end{aligned} \tag{25}$$

where $T = \sigma_k^{(i)^2}$ for $b = +\infty$. This results in an estimate identical to the ML estimate except that the observations will be "clipped" in such a way that in (8b), if the square of the observed value minus the mean exceeds b, the squared value will be replaced by b. In this scheme, b is a constant to be determined from the available information about the deviation of the model; for example, the ϵ in expression (17).

In Fig. 7.3.8, we have illustrated several of the influence functions discussed above. As a comparison, we have also shown the influence function for the ML estimate computed under a generalized Gaussian assumption, i.e., expression (9) with assumed characteristic exponent $\alpha_0 = 0.5, 2.0$, when the data distribution is described by (17) as a deviation from the Gaussian assumption. This influence function can be shown, following the approach in Appendix C, to be given by

$$INF(x, T, f_0) = C(\alpha_0) INF_h(x, T, f_0), \tag{26a}$$

where $INF_h(x, T, f_0)$ is the influence function for the heuristic estimator in expression (24) with β replaced by α_0, while $C(\alpha_0)$ is a constant for a specified α_0, given by

$$C(\alpha_0) = \left[\frac{\Gamma(3/\alpha_0)}{\Gamma(1/\alpha_0)}\right] \alpha_0^{2/\alpha_0 - 1} \left(E_{f_0}\{|Y|_k^{(i)^{\alpha_0}}\}\right)^{2/\alpha_0 - 2}. \tag{26b}$$

Here, $T = \left(E_{f_0}\{|Y|_k^{(i)^{\alpha_0}}\}\right)^{2/\alpha_0}$ and α_0 is the assumed exponent for the generalized Gaussian assumption. For example, with $\alpha_0 = 1$ we find $C(\alpha_0) = 2.0$ indicating that the influence function for the corresponding ML estimator increases twice as fast as that for the statistic of (21), i.e., the special case of the heuristic estimator with $\beta = 1.0$.

It can be seen that, besides the optimal robust estimator, the heuristic estimator is likely to be able to reduce the effects of outliers, while all the estimators are better than the ML under the Gaussian assumption as far as combating outliers are concerned[†] (their influence functions increase slower than that of the ML estimate).

Finally, we want to make two remarks. First of all, the estimators considered here are not all robust in the strict sense of robust statistics, except the optimal robust estimator, in that their influence functions are not bounded. However, the heuristic estimator will greatly reduce the effect of the outliers since its influence function increases much slower as the magnitude of the observation increases. Hence, this scheme may still be effective in practice. Secondly, the heuristic estimator may perform better than the generalized Gaussian model-based ML estimator when $\alpha_0 = \beta$, since its influence function increases slower than that of the generalized Gaussian. For example, we have already seen that when $\alpha_0 = \beta = 1.0$, $C(\alpha_0)$, the ratio of the two influence functions for the generalized Gaussian ML estimator of (9) and the heuristic estimator of (23), is 2.0; for $\alpha_0 = \beta = 0.5$, this ratio increases to 10.1. This means the influence function for the heuristic estimator increases much slower in x than that of the ML estimator under the generalized Gaussian assumption and that this improvement is greater for smaller $\alpha_0 = \beta$. This behavior can also be seen from the results in the next part.

D.) *Application to Synthetic Images*:

As described in Section 7.3.3, the AIC can be applied to determine the number of stochastic classes in an image. This can be achieved by finding the number of clusters in the sample model parameter vectors estimated from a sliding estimation window on

[†] It should be noted that the curves in Fig. 7.3.8 for the Gaussian ML estimate and the generalized Gaussian ML estimate for $\alpha_0 = 0.5$ eventually cross indicating better outlier rejection properties under this α_0 assumption for large deviations.

different spatial positions of the image. When these vectors are obtained, the procedure of identifying the number of classes should be no different from that for the synthetic random data described previously. In addition, the AIC-based model-fitting approach does not put any restriction on the model for the image classes. On the other hand, the parameter vectors obtained from the sliding estimation window might not be a Gaussian mixture or even a generalized Gaussian mixture. If the image classes can be modeled by the independent Gaussian assumption, the estimated mean and the estimated variance will be approximately Gaussian. However, when the image classes are modeled as AR or MRF models, the estimated model parameters need not be Gaussian mixtures. In this case, the robust estimation procedures, described above, may be useful.

In Fig. 7.3.9, we have shown two two-class texture images of size 256 × 256 pixels. In both cases, two textures are combined according to the region map shown in Fig. 7.3.9 a. In the first image (Fig. 7.3.9 b), the textures are realizations of the FQPF AR random field characterized by parameter vector $\mathbf{a}_k = (a_k, b_k, c_k, \sigma_{w_k}), k = 1, 2$, as described previously. In this work, we have chosen $\mathbf{a}_1 = (-0.3, 0.7, 0.5, 9.5), \mathbf{a}_2 = (-0.3, 0.5, 0.7, 14.1)$. A common mean of 100 is added to both classes. In the second image (Fig. 7.3.9 c), the textures are realizations of the BIA MRF's, each characterized by a parameter vector $\mathbf{a}_k = (a_k, b_k), k = 1, 2$. In this work, we have chosen $\mathbf{a}_1 = (-2.0, 1.0), \mathbf{a}_2 = (2.0, -1.0)$.

For each image, the sample model parameter vectors are estimated from a 16 × 16 pixel sliding estimation window with horizontal and vertical displacement, M_2, N_2, also both 16 pixels; hence there are 256 sample mooel parameter vectors for each image. The sample model parameters for the MRF texture image, being two-dimensional, are plotted in Fig. 7.3.10 b. Since the sample model parameter vectors for the AR textures are four-dimensional, they cannot be plotted on a plane; only the the second and third components of these vectors are plotted in Fig. 7.3.10 a. We can see from Fig. 7.3.10, the sample parameter vectors for the MRF texture image do not show a clear clustering tendency; the second and third components of the sample model vectors for the AR textures, however, do show a tendency of two clusters. In Tables 7.3.7 and 7.3.8, we show the computed AIC for the sample model parameter vectors for both images under the assumptions of a generalized Gaussian mixture with $\alpha_0 = 3.0, 2.0, 1.0, 0.5$. The performance, as indicated in

Table 7.3.8 a, for the MRF textures under all assumptions are relatively poor, as expected. The performance indicated in Table 7.3.7 a for the second and third component of the AR textures for all the assumptions of α_0 is encouraging; the AIC makes correct decisions. On the other hand, the AIC computed for all four components of the model vectors of the AR texture image, as indicated in Table 7.3.7 b, is not so encouraging. $K = 2$ provides only a *local* minimum of the AIC's computed. This may due to the fact that the geometrical distribution of the model vectors with four components in the parameter vector space are more complex than that for the case of two components and cause more data points to be outliers. We want to point out, however, the relatively poor performance such as for the MRF does not undermine the efficacy of the AIC but, rather, it points out the need for more careful procedures for obtaining the model parameter vectors. More specifically, the model vectors obtained from sliding estimation windows that contains contaminated data from two texture classes should be detected and rejected such that the remaining set of sample model parameter vectors do show a reasonable cluster tendency. In addition, the success of the AIC on the second and third component of the AR model parameter vectors justifies the efficacy of the AIC in that when the data does show a clustering tendency, the AIC tends to make correct decisions.

An alternative for improving the estimates of the sample model parameter vectors is to use the robust estimator in (23) or the optimal robust variance estimator of (25). The computed AIC using the heuristic estimator with $\beta = 0.5$ and the optimal robust estimator with $\alpha_0 = 2$ for both the AR model parameter vectors (in four dimensions) and the MRF model parameter vectors are shown in Tables 7.3.7 c and 7.3.8 b, respectively. For the optimal robust estimator, the clipper threshold, b, is chosen heuristically to be two or three times the variance estimated under the Gaussian assumption with two clusters since, in this case, very little is known a priori about the model deviation (e.g., ϵ). The results are also shown in Table's 7.3.7 c and 7.3.8 b. In this case, correct decisions are made. However, we see that for the optimal robust estimator to be more practical, some scheme is needed to determine the "clipper threshold", b.

E.) *Application to Real Image Data:*

In this experiment, we attempt to apply the method described in the previous section to estimate the number of statistical image classes in real images. In particular, the images are digitized aerial photographs of size 256×256. For simplicity, we consider the Gaussian model of (13) for the image classes [2]. That is, each image class is modeled by an *i.i.d.* Gaussian random field. Then, each image class can be characterized in terms of a model parameter vector consisting of only two components; the mean and variance.

In Fig.'s 7.3.11 a and 7.3.12 a, we show two aerial photographs. The first contains a building, roads and vegetation while the second contains an oil tank complex surrounded by vegetation. The corresponding computed AIC's for different numbers of clusters are shown in Table 7.3.9 with $K_{max} = 10$. The sliding estimation window is of size 16 x 16 pixels and the vertical and horizontal displacements, M_2 and N_2 in Fig. 7.3.2, are also each 16 pixels. The results suggest that in the first image there are four classes while for the second image five classes best fits the data. The images are segmented using a ML technique [2] with the corresponding model vectors estimated according to that suggested by the AIC criterion and are shown in Fig.'s 7.3.11 and 7.3.12, along with the original images. In these segmentations different tonal areas are well separated. For comparison purpose we have also shown the results of the segmentation using from two up to six classes. It can be seen from the results for both images that, when the assumed number of classes is smaller than that determined by the AIC, a number of significant regions of reasonably large size are missing from the segmentation. On the other hand, when the number of classes is larger than that suggested by the AIC, no significant change in segmentation will result from the increase of the number of classes except the appearance of some noisy regions with small size. This suggests that the AIC model-fitting approach is a reasonable objective approach for practical applications such as image segmentation.

7.3.7 Summary:

In this paper we described a model-fitting approach for determining the number of clusters in observed random data and its applications to stochastic model- based image segmentation. The problem, also known as cluster validation, is solved by finding a best-fitting mixture distribution model to the observed data. The goodness of fit is determined by the AIC criterion. An approximate ML parameter estimation scheme using clustering is proposed to compute the AIC. Experimental results are also described to demonstrate the efficacy, relative robustness and practical applicability of this method.

In the experiments involving synthetic data, the AIC correctly determines the number of clusters in the mixture data, continues to correctly identify the number of data clusters as the distance between the clusters decreases and will, in general, suggest a single cluster when the distance is so small that the data clusters appear visually to be merged. We have investigated the robustness of the AIC under different modeling assumptions using the generalized Gaussian mixture model. In particular, we have used the generalized Gaussian mixture both in providing synthetic data for the experiments and as modeling assumptions in computing the AIC's. The Gaussian mixture assumption is quite robust when the data clusters do not contain many outliers. A generalized Gaussian assumption with $\alpha_0 < 1$ provides very robust performance even when the data clusters do contain many outliers. We have also considered the robustness problem in the light of the theory of robust statistics when the actual data is not a generalized Gaussian mixture, but some deviation from a *nominal* Gaussian mixture. This leads to a heuristic variance estimator and an optimal robust variance estimator for computing the AIC when the data pdf deviates from the nominal Gaussian mixture assumption. This approach makes the performance of the AIC more robust against outliers in the data clusters.

In the application to image data or image segmentation, we have used the AIC to identify the number of image classes where they are modeled as the simple independent Gaussian, or more complex AR and MRF models. The AIC computed using more robust variance estimators, such as the optimal robust variance estimator and the heuristic variance estimator correctly identifies the number of classes in the synthetic AR or MRF mixture images. For real aerial photo images, where the true number of classes is unknown, the AIC computed based on a Gaussian mixture assumption provides identification results that agree well with *subjective* assessments.

This work also brings up several interesting issues for further investigation. For example, it would be of interest to use the EM algorithm as the estimation method for computing the AIC and compare the results with those described in this paper. Another interesting issue is to further understand the theoretical aspects of the minimum AIC principle; for example, to characterize the performance of the AIC under different data distributions in terms of probability of correct decision. Finally, additional study is required on the application of the AIC to image segmentation. More specifically, estimation methods should be developed to reject sample model parameter vectors from sliding estimation windows that contain a significant amount of data from more than one image class; or as an alternative, robust estimation techniques, such as the ones described in this paper, need to be further tested that will diminish the effects of such "contaminated" sample model parameter vectors.

Appendix A

Maximum-Likelihood Estimation for Generalized Gaussian Models

In this section, we will derive the ML estimate of the variance of a generalized pdf model from a set of observations. For simplicity, we consider one-dimensional observations and assume that the mean of the pdf is zero. Let $\mathbf{x} = \{x_1, x_2, \dots x_N\}$ (here $\mathbf{x}$ is *not* used for images as in Section 7.3.3) be a set of *i.i.d.* observations of a random variable X with a generalized Gaussian pdf

$$p(x) = \frac{\alpha\eta}{2\Gamma(1/\alpha)} exp[-|\eta x|^{\alpha}], \tag{A1}$$

where

$$\eta = \frac{1}{\sigma}\left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right]^{1/2}, \tag{A2}$$

and σ^2 is the variance. Assume that $\alpha > 0$ is known; we would like to estimate σ^2 based on the observation $\mathbf{x}$. From equation (A2), this is equivalent to estimating η. An ML estimate of η, denoted by $\hat{\eta}_{ML}$, is one that maximizes the *likelihood* functional with respect to η for given $\mathbf{x}$. That is

$$\hat{\eta}_{ML} = \arg\max_{\eta} p_{\mathbf{x}}(\mathbf{x}|\eta), \tag{A3}$$

where

$$p_{\mathbf{x}}(\mathbf{x}|\eta) = \prod_{i=1}^{N} p(x_i|\eta), \tag{A4}$$

and the $p(x_i|\eta)$'s are as in (A1). Maximizing (A3) is equivalent to maximizing the logarithm of (A4), i.e.,

$$\begin{aligned} log\{p_{\mathbf{x}}(\mathbf{x}|\eta)\} &= \sum_{i=1}^{N} log[p(x_i|\eta)] \\ &= N\left[log\eta + log(\frac{\alpha}{2\Gamma(1/\alpha)})\right] - (\sum_{i=1}^{N} \eta^{\alpha}|x_i|^{\alpha}). \end{aligned} \tag{A5}$$

Taking the derivative of (A5) with respect to η, setting it to zero and solving for η, we have

$$\frac{N}{\eta} - \alpha\eta^{\alpha-1}(\sum_{i=1}^{N}|x_i|^{\alpha}) = 0, \tag{A6}$$

and thus

$$\hat{\eta}_{ML}^{-1} = \left(\frac{\alpha}{N}\sum_{i=1}^{N}|x_i|^{\alpha}\right)^{1/\alpha}. \tag{A7}$$

Finally,

$$\hat{\sigma}_{ML} = \hat{\eta}_{ML}^{-1}\left[\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}\right]^{1/2}. \tag{A8}$$

Appendix B

The Expressions for the AIC Under Different Modeling Assumptions

In this section, we will show that, under the Gaussian or the generalized Gaussian mixture assumption for the individual components, the *log* maximum likelihood in the expression for the AIC will only depend on the *the number of samples* in each cluster and the *intra-cluster variances* for each cluster.

We still use the notation established in Section 7.3.3. Suppose that the observed data can be represented by N *i.i.d.* random vectors, $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$. In [1], we have shown for the approximate ML parameter estimation method which first performs a clustering, the likelihood functional is[†]

$$p_K(\mathbf{Y}) = \prod_{k=1}^{K} \pi_k^{N_k} \prod_{j=1}^{N_k} p_k(\mathbf{y}_{k_j}). \tag{B1}$$

Taking the logarithm,

$$log\{p_K(\mathbf{Y})\} = \sum_{k=1}^{K} \left(N_k log \pi_k + \sum_{j=1}^{N_k} log p_k(\mathbf{y}_{k_j}) \right). \tag{B2}$$

Assume the $\mathbf{y}$'s, $\mathbf{y} \epsilon \mathbf{Y}$, are m-dimensional with independent components, then

$$log\{p_K(\mathbf{Y})\} = \sum_{k=1}^{K} \left(N_k log \pi_k + \sum_{j=1}^{N_k} \sum_{i=1}^{m} log p_k^{(i)}(y_{k_j}^{(i)}) \right), \tag{B3}$$

where $p_k^{(i)}(\cdot)$ is the i'th component pdf of the k'th class of the random data. It is easily seen that the ML estimates (for convenience, we suppress the subscript ML for all the ML estimates) of the mixture weights, π_k, are given by

$$\hat{\pi}_k = \frac{N_k}{N}; k = 1, 2, ..., K. \tag{B4}$$

Replacing the model parameters in (B3) by their ML estimates then yields

[†] For notational convenience, we have suppressed the functional dependence upon **a** in writing $p_k(\mathbf{Y})$ for $p_k(\mathbf{Y}|\mathbf{a})$ as in (11).

$$log\{p_K(\mathbf{Y})\} = \sum_{k=1}^{K} \left(N_k log \left(\frac{N_k}{N} \right) + \sum_{j=1}^{N_k} \sum_{i=1}^{m} log \hat{p}_k^{(i)}(y_{k_j}^{(i)}) \right), \tag{B5}$$

where $\hat{p}_k^{(i)}(y_{k_j}^{(i)})$ is the pdf $p_k^{(i)}(y_{k_j}^{(i)})$ with all parameters replaced by their corresponding ML estimates. Now, we need to find the second term in the brackets under different modeling assumptions.

1.) *The Gaussian Mixture:*

In the case of Gaussian mixture with ML estimates,

$$log\{\hat{p}_k^{(i)}(y_{k_j}^{(i)})\} = \frac{1}{\sqrt{2\pi}\hat{\sigma}_k^{(i)}} exp\left[- \frac{(y_{k_j}^{(i)} - \hat{m}_k^{(i)})^2}{2\hat{\sigma}_k^{(i)^2}} \right]. \tag{B6}$$

Then

$$\begin{aligned} &\sum_{j=1}^{N_k} \sum_{i=1}^{m} log \hat{p}_k^{(i)}(y_{k_j}^{(i)}) \\ &= - \sum_{j=1}^{N_k} \sum_{i=1}^{m} \left[\frac{1}{2} log 2\pi \hat{\sigma}_k^{(i)^2} + \frac{(y_{k_j}^{(i)} - \hat{m}_k^{(i)})^2}{2\hat{\sigma}_k^{(i)^2}} \right] \\ &= -\frac{1}{2} m N_k log 2\pi - N_k \sum_{i=1}^{m} log \hat{\sigma}_k^{(i)} - \sum_{j=1}^{N_k} \sum_{i=1}^{m} \frac{(y_{k_j}^{(i)} - \hat{m}_k^{(i)})^2}{2\hat{\sigma}_k^{(i)^2}} \\ &= -\frac{1}{2} m N_k log 2\pi - N_k \sum_{i=1}^{m} log \hat{\sigma}_k^{(i)} - \frac{m N_k}{2}. \end{aligned} \tag{B7}$$

If we substitute (B7) in (B5) and drop all the terms that do not depend on the data or the order, K, of the mixture model, we will arrive at an *equivalent* log-likelihood functional, or sufficient statistic, given by

$$L_K(\mathbf{Y}) = \sum_{k=1}^{K} N_k \big(log N_k - \sum_{i=1}^{m} log \hat{\sigma}_k^{(i)} \big). \tag{B8}$$

It is indeed dependent only on the intra-cluster variance and the number of samples in the clusters. This quantity can then be used in place of the log-likelihood functional in computing $AIC(K)$ according to (3).

2.) *The Generalized Gaussian Mixture*:

The generalized Gaussian distribution has a similar exponential structure as the Gaussian distribution. Hence, the derivation here is similar to the previous part. More specifically,

$$\hat{p}_k^{(i)}(y_{k_j}^{(i)}) = \frac{\alpha\hat{\eta}_k^{(i)}}{2\Gamma(1/\alpha)} exp\left[- |\hat{\eta}_k^{(i)}(y_{k_j}^{(i)} - \hat{m}_k^{(i)})|^{\alpha}\right]. \tag{B9}$$

Then, similar to the Gaussian case, we have

$$\begin{aligned} &\sum_{j=1}^{N_k}\sum_{i=1}^{m} log\hat{p}_k^{(i)}(y_{k_j}^{(i)}) \\ &= \sum_{j=1}^{N_k}\sum_{i=1}^{m}\left(log\hat{\eta}_k^{(i)} - log(\frac{\alpha}{2\Gamma(1/\alpha)}) - |\hat{\eta}_k^{(i)}(y_{k_j}^{(i)} - \hat{m}_k^{(i)})|^{\alpha}\right) \\ &= -N_k\sum_{i=1}^{m} log\left(\frac{1}{\hat{\eta}_k^{(i)}}\right) - mN_k log(\frac{\alpha}{2\Gamma(1/\alpha)}) - \frac{mN_k}{\alpha}. \end{aligned} \tag{B10}$$

Substituting this in (B5) and dropping the terms that do not depend on the data, we have

$$L_K(\mathbf{Y}) = \sum_{k=1}^{K} N_k\left[logN_k - \sum_{i=1}^{m} log\left(\frac{1}{\hat{\eta}_k^{(i)}}\right)\right]. \tag{B11}$$

Notice that $1/\hat{\eta}_k^{(i)}$ is proportional to the square root of the cluster component variance $\hat{\sigma}_k^{(i)}$. Hence, (B11) is of the same form as (B8) when we again eliminate data independent terms.

Appendix C

Examples of Influence Functions

In this section, we will derive the influence functions for the estimators outlined in Section 7.3.4-C. For simplicity, assume that the observed data are *i.i.d.* random variables, denoted by $Y_1, Y_2, \ldots, Y_n$. Assume the ideal pdf, f_0, of the random variables is zero-mean, unit-variance Gaussian. Then the influence function of the asymptotic limit of an estimate, denoted by T, is defined as in (19) of Section 7.3.4-C, by

$$INF(x, T, f_0) = \frac{\partial}{\partial t}[T((1-t)f_0 + t\delta_x)] \,|_{t=0} \,. \tag{C1}$$

Here, x is a contaminated observation, T is considered as a functional of the underlying data distribution, and δ_x is a delta-function pdf at x. For the examples we have considered throughout this paper, T_n, the estimate of a statistic based on n samples is always related to an ***averaging*** operator (e.g., (8), (9), (18), (23)); hence, T is always related to an ***expectation*** operator. For example, for the heuristic variance estimator of (23) in Section 7.3.4-C,

$$\hat{\sigma}_n^2 = \left[\frac{1}{n}\sum_{i=1}^{n} |y_i|^\beta\right]^2, \tag{C2}$$

where $0 < \beta \leq 1$. In this case, $T_n = \hat{\sigma}_n^2$, hence,

$$T(f) = E_f^2\{|Y|^\beta\}, \tag{C3}$$

where f is any valid pdf of Y, the random variable.

To find the influence function ᶜᶠ the heuristic variance estimator, we can take the partial derivative of T in (C3) with respect to t, where $f = (1-t)f_0 + t\delta_x$. Therefoıe,

$$\begin{aligned} &\frac{\partial}{\partial t}[T((1-t)f_0 + t\delta_x)] \\ &= 2(E_f\{|Y|^\beta\})(-E_{f_0}\{|Y|^\beta\} + E_{\delta_x}\{|Y|^\beta\}) \\ &= 2(E_f\{|Y|^\beta\})(-E_{f_0}\{|Y|^\beta\} + |x|^\beta). \end{aligned} \tag{C4}$$

Here, we have used the following property of the expectation operator

$$E_{(1-t)f_1+tf_2}\{Y\} = (1-t)E_{f_1}\{Y\} + tE_{f_2}\{Y\}, \tag{C5a}$$

for pdf's f_1 and f_2, and $0 \leq t \leq 1$. We have also used the property of the delta-function to obtain

$$E_{\delta_x}\{Y\} = x. \tag{C5b}$$

Finally, let $t = 0$ in (C4), the influence function for the heuristic variance estimator is

$$INF_h(x, T, f_0) = 2\big(E_{f_0}\{|Y|^\beta\}\big)\big(|x|^\beta - E_{f_0}\{|Y|^\beta\}\big), \tag{C6}$$

which is the same as (25) of Section III. Other influence functions discussed in this paper can be found using the same procedure as presented above.

References for Section 7.3

[1] B. S. Everitt, *Cluster Analysis*, published by Heinemann Education Books (for) Social Science Research Council, London.

[2] J. Zhang and J. W. Modestino, "Image Segmentation Using a Gaussian Model", Technical Report, ECSE Dept., RPI, March 1987.

[3] J. Zhang and J. W. Modestino, " Unsupervised AR Random Field Model-Based Image Segmentation", Technical Report, ECSE Dept., RPI, March 1987.

[4] J. Zhang and J. W. Modestino, "Texture Classification and Discrimination Using the Markov Random Field Model", submitted to Pattern Anal. and Mach. Intell., Sept. 1987.

[5] K. Fukunaka, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[6] J. T. Tou and R. C. Gonzalez, *Pattern Recognition Principles*, Addison-Wesley Publishing Company, Reading, MA, 1974.

[7] A. K. Jain,"Cluster Analysis", *Handbook of Pattern Recognition and Image Processing*, Ed. by T.Y. Young and K.S. Fu, Academic Press, New York, 1986.

[8] R. Dubes and A. K. Jain, "Validity Studies in Clustering Methodologies", *Pattern Recognition*, Vol. 11, pp. 235-254, 1979.

[9] D. M. Titterington, A.F.M. Smith and U.E. Makov, *Statistical Analysis of Finite Mixture Distributions*, John Wiley & Sons, New York, 1985.

[10] H. Akaike, "A New Look at the Statistical Model Identification", IEEE Trans. Automatic Control, AC-19, pp. 716-722, Dec. 1974.

[11] H. Akaike, "An Entropy Maximization Principle", Proc. Symp. on *Applied Statistics*, Ed. P. R. Krishnaiah, pp. 27-41, North Holland, Amsterdam, 1977.

[12] H. Akaike, "A Bayesian Extension of the Minimum AIC Procedure", Biometrika, Vol. 66, pp. 237-242, 1979.

[13] H. Akaike, "A Bayesian Analysis of the Minimum AIC Procedure", Ann. Inst. Statis. Math., Vol. 30A, pp. 9-14, 1979.

[14] H. Akaike, "Canonical Correlation Analysis of Time Series and the Use of an Information Criterion", in *System Identification Advances and Case Studies*, Ed. by R. K.

Mehra and D. G. Lainiotis, Academic Press, New York, 1976.

[15] G. Schwarz, "Estimating the Dimension of a Model", The Ann. of Statistics, Vol. 6, No. 2, pp. 461-464, 1978.

[16] R. L. Kashyap, R. Chellappa and N. Ahuja, "Decision Rules for Choice of Neighbors in Random Field Models of Images",*Computer Graphics and Image Processing*, Vol. 15, pp. 301-318, 1981.

[17] S. L. Sclove, "Application of the Conditional Population-Mixture Model to Image Segmentation", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-5, pp. 428-433, July 1983.

[18] H. L. Van Trees, *Detection, Estimation, and Modulation Theory: Part I*, John Wiley & Sons, New York, 1968.

[19] P. J. Huber, "The Behavior of Maximum Likelihood Estimates Under Non-Standard Conditions", Proc. 5th Berkeley Symp. on Math. Stat. and Probability, Vol. 1, pp. 221-233, 1967.

[20] J. Zhang, "Statistical Model-Based Image Analysis", Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, New York, August 1988.

[21] A. K. Jain and J. V. Moreau, "Bootstrap technique in cluster analysis", Pattern Recognition, Vol. 20, No.5, pp. 547-568, 1987.

[22] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data-set", Psychometrika, 50, pp. 159-175, 1985.

[23] D. M. Titterington, "Comments on 'application of the conditional population-mixture model to image segmentation' ", IEEE Trans. Pattern Anal. Machine Intel., PAMI-6, pp. 656-657, Nov. 1984.

[24] R. A. Redner and H. F. Walker, "Mixture densities, maximum- likelihood and the EM algorithm", SIAM Rev., Vol. 26, pp. 195-239, 1984.

[25] J. Rissanen, "Modeling by shortest data description", Automatica, Vol. 14, pp. 465-471, 1978.

[26] J. Segen and A. C. Sanderson, "Model inference and pattern discovery by minimal representation method", Technical Report, Carnegie-Mellon University, CMU-RI-TR-82-2, 1982.

[27] J. Zhang and J. W. Modestino, "A model-fitting approach to cluster validation with applications to stochastic model-based image segmentation", *Proc. ICASSP '88*, New York, April 1988.

[28] J. Besag, "Spatial interaction and the statistical analysis of lattice system (with discussion)", J. of Royal Statist. Soc., Series B, Vol. 36, pp. 312-326, 1974.

[29] R. Kinderman and J. L. Snell, *Markov Random Fields and Their Applications*, Providence: Amer. Math. Soc., 1980.

[30] B. M. Prestley, *Spectrum Analysis and Time Series*, Academic Press, New York, 1982.

[31] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-6, pp.721-741. Nov. 1984.

[32] C. W. Therrien, T. F. Quatieri and D. E. Dudgeon,"Statistical model-based algorithms for image analysis", Proc. IEEE, Vol. 74, pp. 532-551, April 1986.

[33] H. Derin and H. Elliot, "Modeling and segmentation of noisy and textured images using Gibbs random fields", IEEE Trans. Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 39-55, January 1987.

[34] F. S. Cohen and D. B. Cooper, "Simple, parallel, hierachical and relaxation algorithms for segmenting non-casual Markovian random field models", Proc. IEEE Pattern Anal. Machine Intel., Vol. PAMI-9, pp. 195-219, March 1987.

[35] J. Besag, "On the statistical analysis of dirty pictures", J. Royal Stat. Soc. B., Vol. 48, pp. 259-302, 1986.

[36] J. Marroquin, S. Mitter, and T. Poggio, "Computational Vision", J. of Amer. Stat. Assc., Vol. 82, pp. 76-89.

[37] J. Zhang and J. W. Modestino, "A MRF model-based approach for image interpretation", in preparation.

[38] P. J. Huber, *Robust Statistics*, John Wiley and Sons, Inc., New York, 1981.

[39] F. R. Hampel, P. J. Rousseeuw, E. M. Ronchetti, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, John Wiley and Sons, Inc., New York, 1986.

[40] S. A. Kassam and H. V. Poor,"Robust techniques for signal processing: a survey",

Proc. IEEE, Vol. 73, pp. 433-481, March 1985.

[41] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall Inc., 1988.

[42] D. M. Hawkins, M. W. Muller, and J. A. Ten Krooden, "Cluster analysis", in *Topics in Applied Multivariate Analysis*, Cambridge University Press, Cambridge, pp. 303-356, 1982.

[43] M. Aitkin and D. B. Rubin, "Estimation and hypothesis testing in finite mixture models", J. Royal Statist. Soc. Series B., Vol. 47, pp. 67-75, 1985.

[44] B. P. Everitt and D. J. Hand, *Finite Mixture Distributions*, Chapman and Hall, London and New York, 1981.

K	AIC (K)
1	998
2	**388** (min)
3	463
4	472
5	490
6	549
7	580
8	557

a.) Two-component Gaussian mixture

K	AIC (K)
1	960
2	712
3	**586** (min)
4	617
5	681
6	703
7	692
8	709

b.) Three-component Gaussian mixture

K	AIC (K)
1	988
2	852
3	805
4	**717** (min)
5	753
6	782
7	806
8	803

c.) Four-component Gaussian mixture

Table 7.3.2

Computed AIC's for the Synthetic Data with K_{max}=8.

	Number of Clusters	Assumed Exponential Parameter			
Intercluster Distance	K	$\alpha_o = 3.0$	$\alpha_o = 2.0$	$\alpha_o = 1.0$	$\alpha_o = 0.5$
r = 5.0	1	1383	1310	1196	1105
	2	1186	1121	1033	972
	3	669 (min)	544 (min)	352 (min)	205 (min)
	4	713	582	385	239
	5	738	602	403	258
r = 2.5	1	820	719	563	442
	2	887	773	601	472
	3	661 (min)	537 (min)	346 (min)	201 (min)
	4	683	558	371	232
	5	750	630	447	310
r = 2.0	1	671	560	385	246
	2	766	649	470	331
	3	628 (min)	505 (min)	316 (min)	170 (min)
	4	676	556	379	249
	5	681	562	385	252
r = 1.5	1	508 (min)	386 (min)	196 (min)	49 (min)
	2	600	480	299	162
	3	556	432	244	96
	4	598	482	315	188
	5	596	478	306	175

Table 7.3.3

Computed AIC's for the Generalized Gaussian Mixture Data with $\alpha = 3.0$.

	Number of Clusters	Assumed Exponential Parameter			
Intercluster Distance	K	$\alpha_o = 3.0$	$\alpha_o = 2.0$	$\alpha_o = 1.0$	$\alpha_o = 0.5$
r = 5.0	1	1374	1300	1186	1095
	2	1168	1095	994	925
	3	696 (min)	542 (min)	316 (min)	150 (min)
	4	739	587	366	208
	5	776	619	400	247
r = 2.5	1	805	703	547	426
	2	856	728	556	436
	3	668 (min)	519 (min)	298 (min)	134 (min)
	4	702	556	343	187
	5	744	599	390	239
r = 2.0	1	656	542	369	236
	2	780	650	464	327
	3	625 (min)	484 (min)	274 (min)	117 (min)
	4	681	529	333	182
	5	701	563	364	215
r = 1.5	1	494 (min)	505 (min)	177 (min)	33 (min)
	2	641	505	307	158
	3	552	411	207	52
	4	590	452	262	127
	5	615	478	278	126

Table 7.3.4

Computed AIC's for the Generalized Gaussian Mixture with $\alpha = 2.0$ (the Gaussian Mixture).

	Number of Clusters	Assumed Exponential Parameter			
Intercluster Distance	K	$\alpha_o = 3.0$	$\alpha_o = 2.0$	$\alpha_o = 1.0$	$\alpha_o = 0.5$
r = 5.0	1	1376	1356	1196	1113
	2	1184	1092	965	880
	3	718 (min)	494 (min)	167 (min)	-56 (min)
	4	725	519	216	6
	5	776	576	293	97
r = 2.5	1	899	707	559	447
	2	855	724	542	414
	3	656 (min)	450 (min)	146 (min)	-69 (min)
	4	664	473	193	-6
	5	726	531	258	72
r = 2.0	1	662	544	379	257
	2	742	604	410	271
	3	614 (min)	413 (min)	117 (min)	-93 (min)
	4	635	442	164	56
	5	692	499	235	56
r = 1.5	1	506 (min)	366	174	37
	2	632	475	257	103
	3	561	359 (min)	77 (min)	-120 (min)
	4	605	403	129	-56
	5	652	456	187	-2

Table 7.3.5

Computed AIC's for the Generalized Gaussian Mixture with $\alpha = 1.0$ (the Laplacian Mixture).

	Number of Clusters	Assumed Exponential Parameter			
Intercluster Distance	K	α_o = 3.0	α_o = 2.0	α_o = 1.0	α_o = 0.5
r = 5.0	1	1370	1299	1195	1120
	2	1190	1078	914	802
	3	784	448	-64 (min)	-41 (min)
	4	694 (min)	404 (min)	-48	-364
	5	707	419	-18	-317
r = 2.5	1	807	697	556	458
	2	851	698	483	329
	3	656	358	-102 (min)	-432 (min)
	4	633 (min)	355 (min)	-66	-364
	5	600	378	-26	-293
r = 2.0	1	668	533	370	261
	2	744	572	343	183
	3	642	334	-120	-433 (min)
	4	556 (min)	282 (min)	-123 (min)	-404
	5	578	311	-72	-303
r = 1.5	1	531 (min)	353	150	21
	2	641	432	179	20
	3	625	309	-126 (min)	-412 (min)
	4	619	333	-29	-255
	5	564	289 (min)	-77	-312

Table 7.3.6

Computed AIC's for the Generalized Gaussian Mixture with α = 0.5.

K	$\alpha_o = 3.0$	$\alpha_o = 2.0$	$\alpha_o = 1.0$	$\alpha_o = 0.5$
1	-109	-115	-122	-128
2	-122 (min)	-130 (min)	-142 (min)	-150 (min)
3	-120	-127	-138	-146
4	-118	-125	-135	-143
5	-116	-123	-134	-142

a) AIC's computed for the second and third components of the AR model vector.

K	$\alpha_o = 3.0$	$\alpha_o = 2.0$	$\alpha_o = 1.0$	$\alpha_o = 0.5$
1	-155	-166	-181	-191
2	-190	-207	-230	-248
3	-189	-206	-229	-245
4	-191	-208	-231	-247
5	-192 (min)	-209 (min)	-234 (min)	-250 (min)

b) AIC's computed for all four components of the AR model vectors.

K	Heuristic ($\beta = 0.5$)	Robust ($\alpha_o = 2.0$)
1	-939	-1978
2	-1144 (min)	-2179 (min)
3	-1075	-2149
4	-1057	-2153
5	-1032	-2136

c) AIC's computed with the heuristic variance estimator with $\beta = 0.5$ and the robust estimator with $\alpha_o = 2.0$

Table 7.3.7

Computed AIC's for the Two-Class AR Texture Image

K	$\alpha_o = 3.0$	$\alpha_o = 2.0$	$\alpha_o = 1.0$	$\alpha_o = 0.5$
1	196	161	114	78
2	-81	-174	-304	-393
3	-77	-175	310	-404
4	-151 (min)	-240 (min)	-367 (min)	-457 (min)
5	-141	-232	-358	-445

a) AIC's computed based on generalized Gaussian assumptions.

K	Heuristic ($\beta = 0.5$)	Robust ($\alpha_o = 2.0$)
1	41	-310
2	-106 (min)	-331 (min)
3	-62	-287
4	-39	-290
5	-4	-274

b) AIC's computed with the heuristic and robust variance estimator.

Table 7.3.8

AIC's Computed for the Two-Class MRF Texture Images.

K	AIC (K)
1	526
2	534
3	511
4	**506** (min)
5	515
6	518
7	519
8	512
9	518
10	526

a.) Road Scene

K	AIC (K)
1	882
2	768
3	705
4	679
5	**664** (min)
6	682
7	674
8	677
9	670
10	672

b.) Oil Tank Scene

Table 7.3.9

Computed AIC's for the Real Image Data with K_{max}=10.

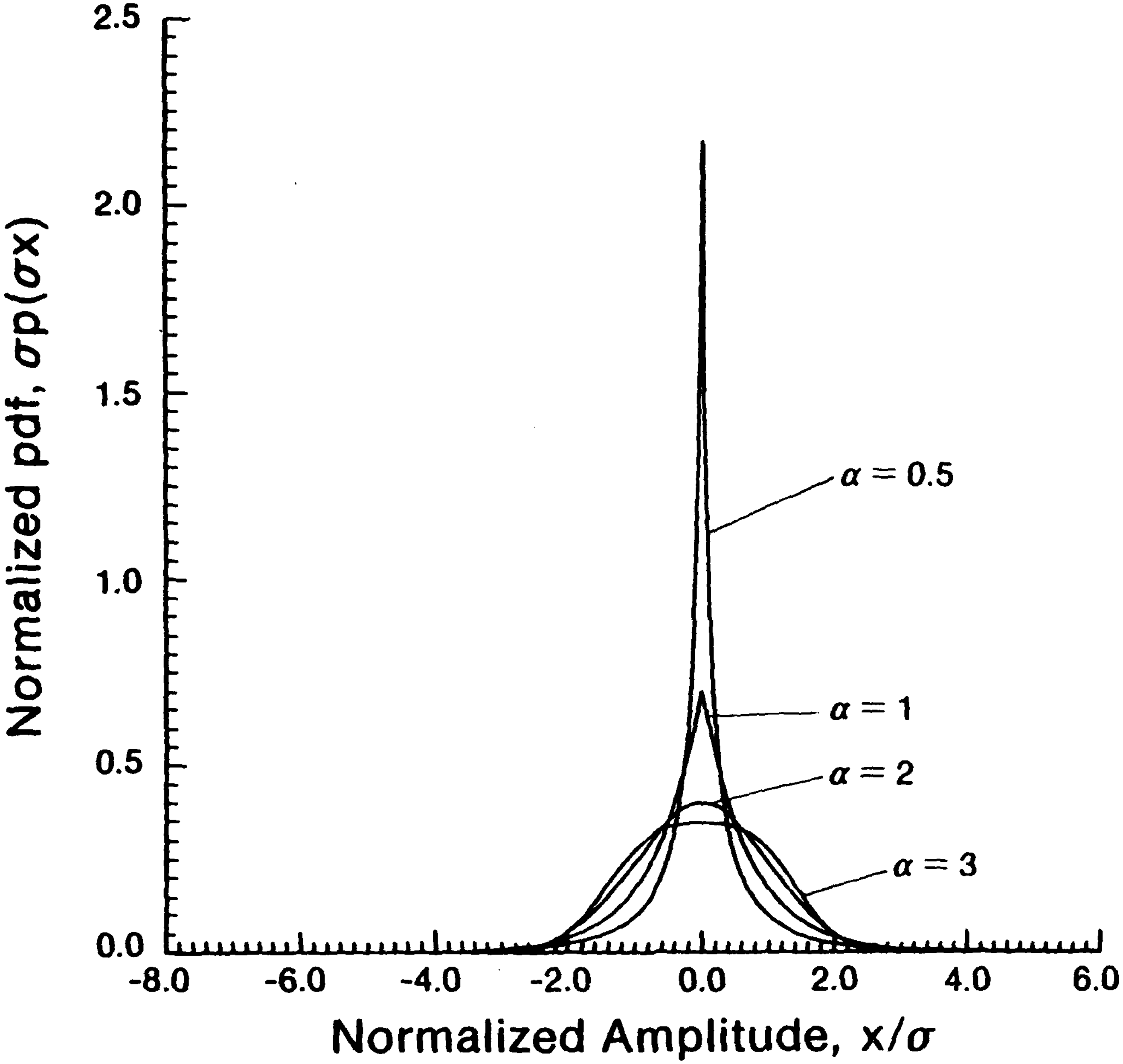


Figure 7.3.1

Examples of Normalized Generalized Gaussian Distribution

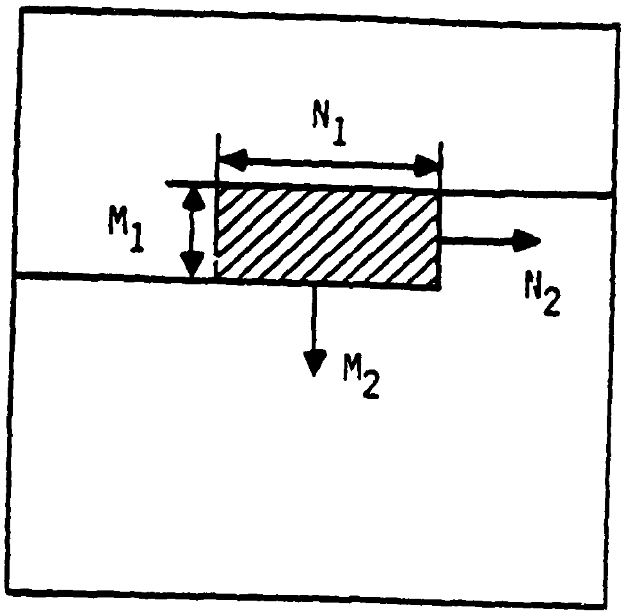
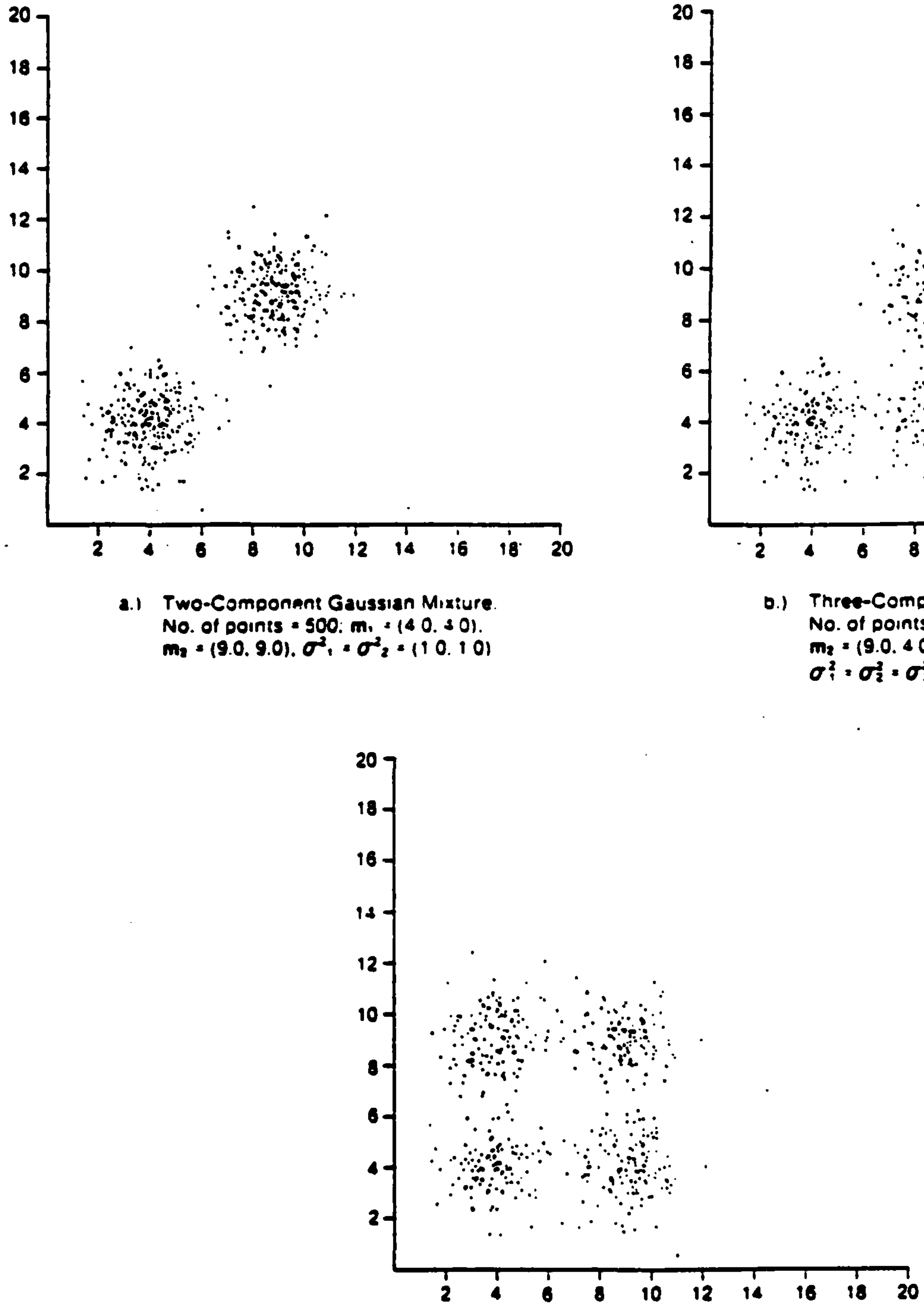


Figure 7.3.2

The Sliding Estimation Windcw.



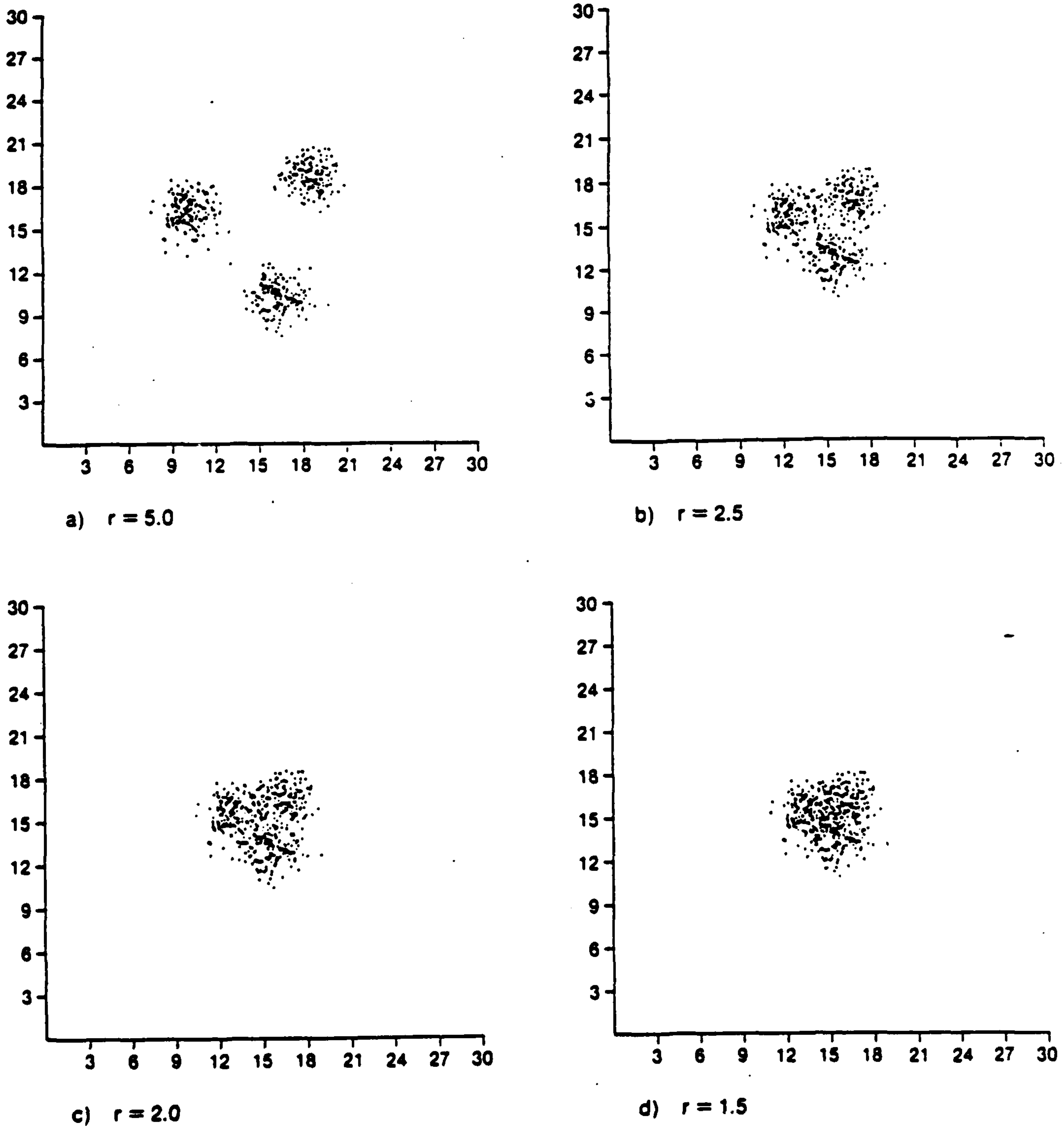
a.) Two-Component Gaussian Mixture.
No. of points = 500; m_1 = (4.0, 4.0).
m_2 = (9.0, 9.0), $\sigma^2_1 = \sigma^2_2$ = (1.0, 1.0)

b.) Three-Component Gaussian Mixture.
No. of points = 500; m_1 = (4.0, 4.0),
m_2 = (9.0, 4.0), m_3 (9.0, 9.0),
$\sigma^2_1 = \sigma^2_2 = \sigma^2_3$ = (1.0, 1.0).

c.) Four-Component Gaussian Mixture.
No of points = 500; m_1 = (4.0, 4.0).
m_2 (9.0, 4.0), m_3 (4.0, 9.0)
m_4 (9.0, 9.0), $\sigma^2_1 = \sigma^2_2 = \sigma^2_3 = \sigma^2_4$ = (1.0, 1.0).

Figure 7.3.3

Examples of Synthetic Gaussian Mixture Data.



a) r = 5.0

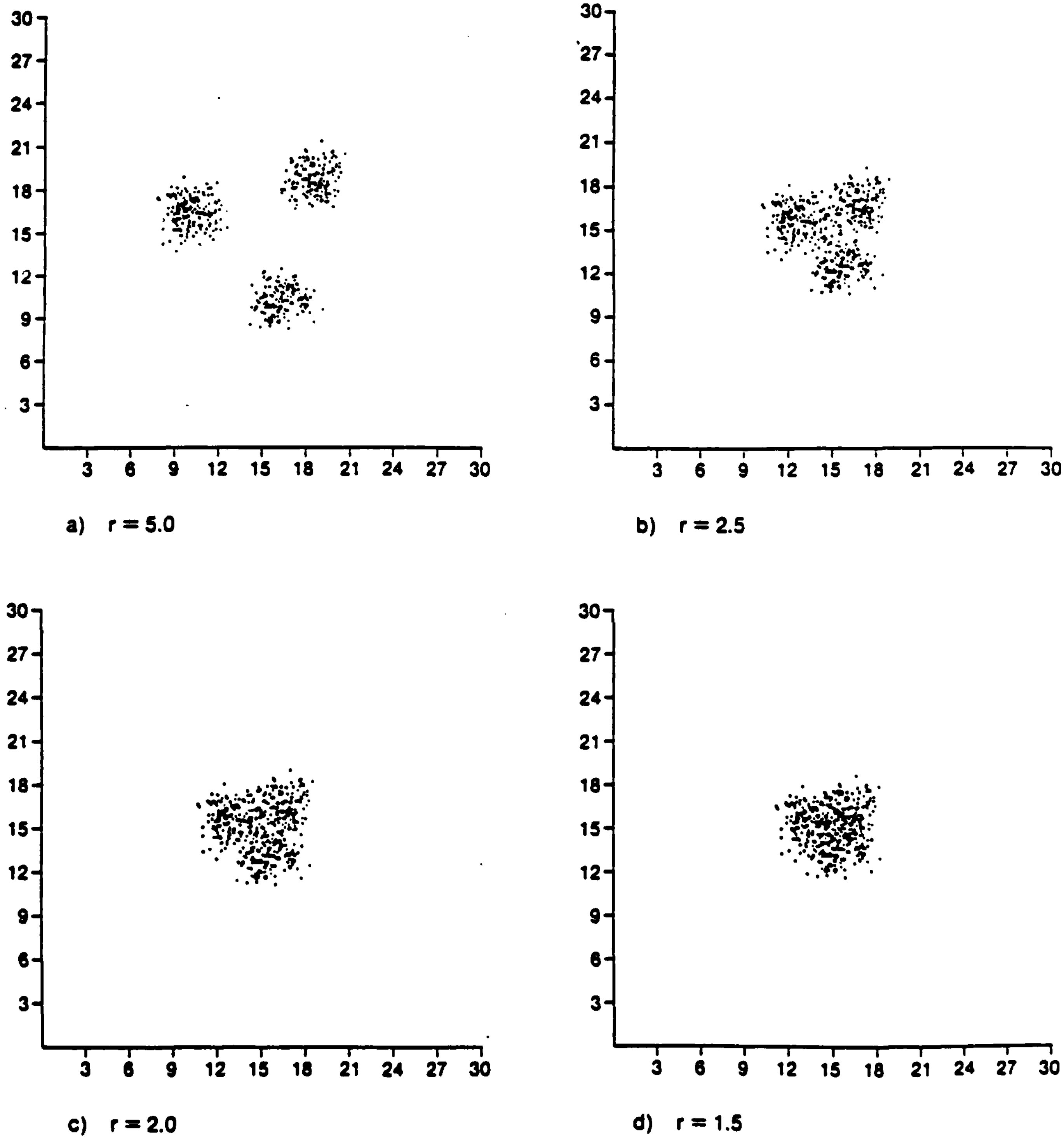
b) r = 2.5

c) r = 2.0

d) r = 1.5

Figure 7.3.4

Generalized Gaussian Mixtures with α=3.0, σ^2=1.0



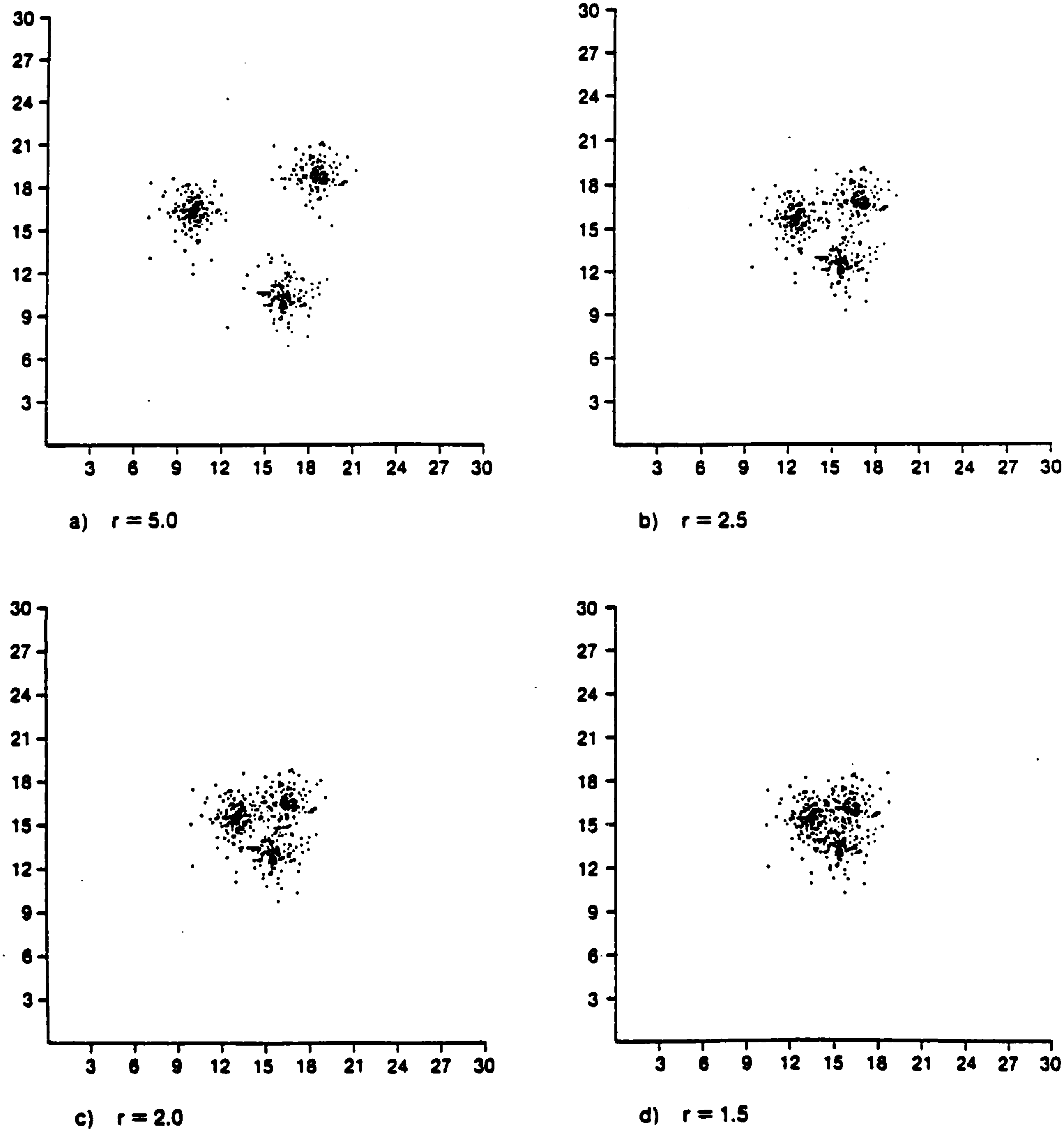
a) r = 5.0

b) r = 2.5

c) r = 2.0

d) r = 1.5

Figure 7.3.5
Generalized Gaussian Mixtures with α=2.0, σ^2=1.0



a) r = 5.0

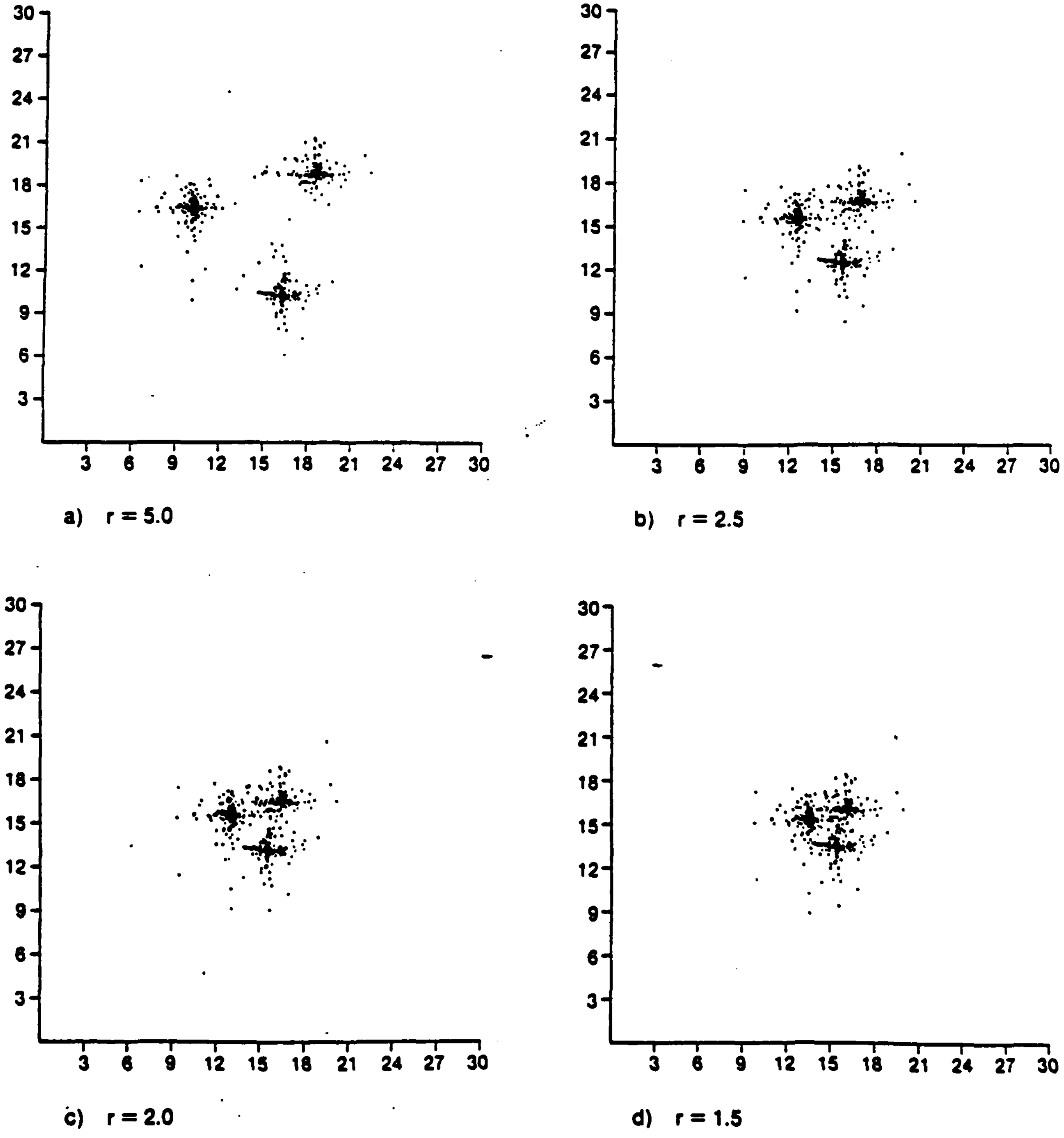
b) r = 2.5

c) r = 2.0

d) r = 1.5

Figure 7.3.6

Generalized Gaussian Mixtures with α=1.0, σ^2=1.0



a) r = 5.0

b) r = 2.5

c) r = 2.0

d) r = 1.5

Figure 7.3.7

Generalized Gaussian Mixtures with α=0.5, σ^2=1.0

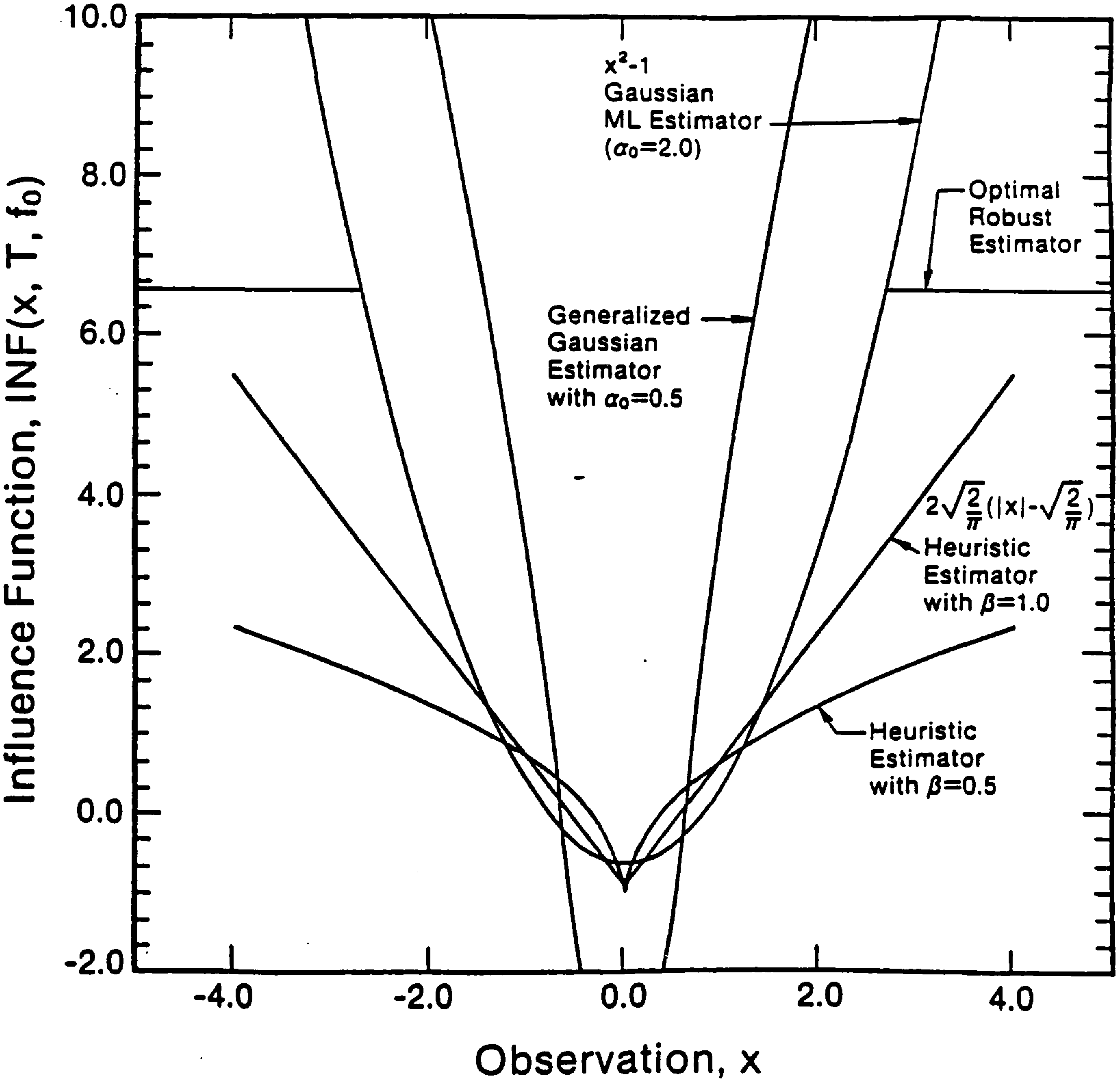
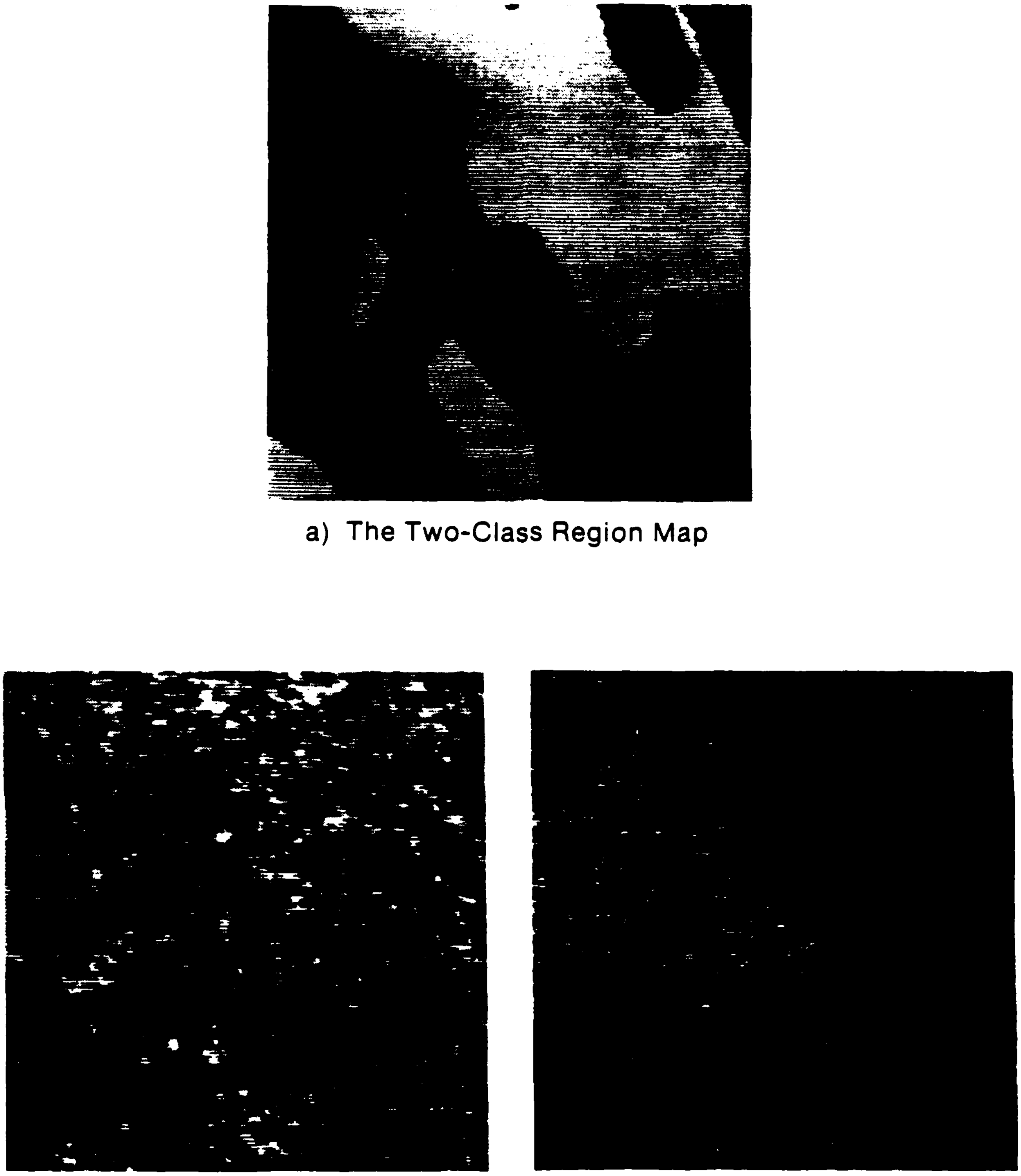


Figure 7.3.8

Examples of Influence Functions for Estimation of Intraclass Spread



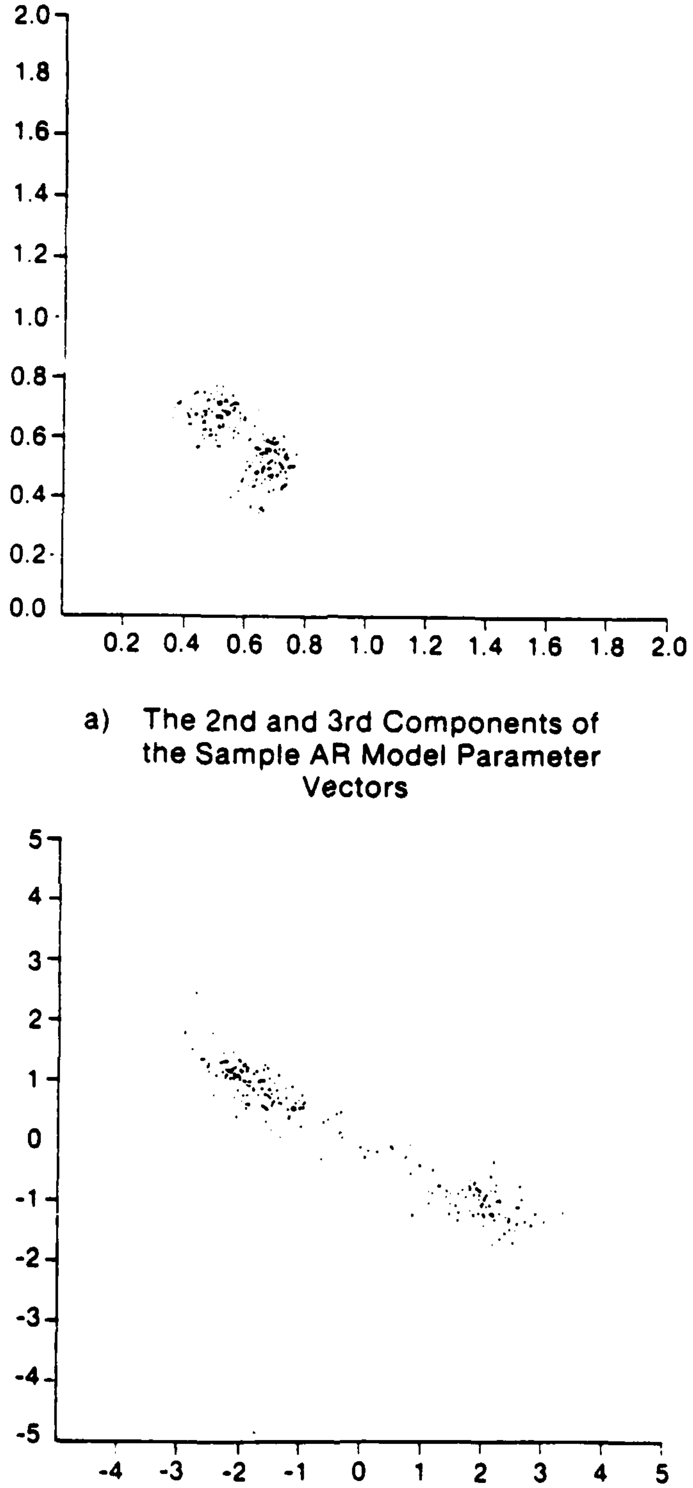
a) The Two-Class Region Map

b) The Two-Class AR Texture Image

c) The Two-Class MRF Texture Image

Figure 7.3.9

Example of Texture Images

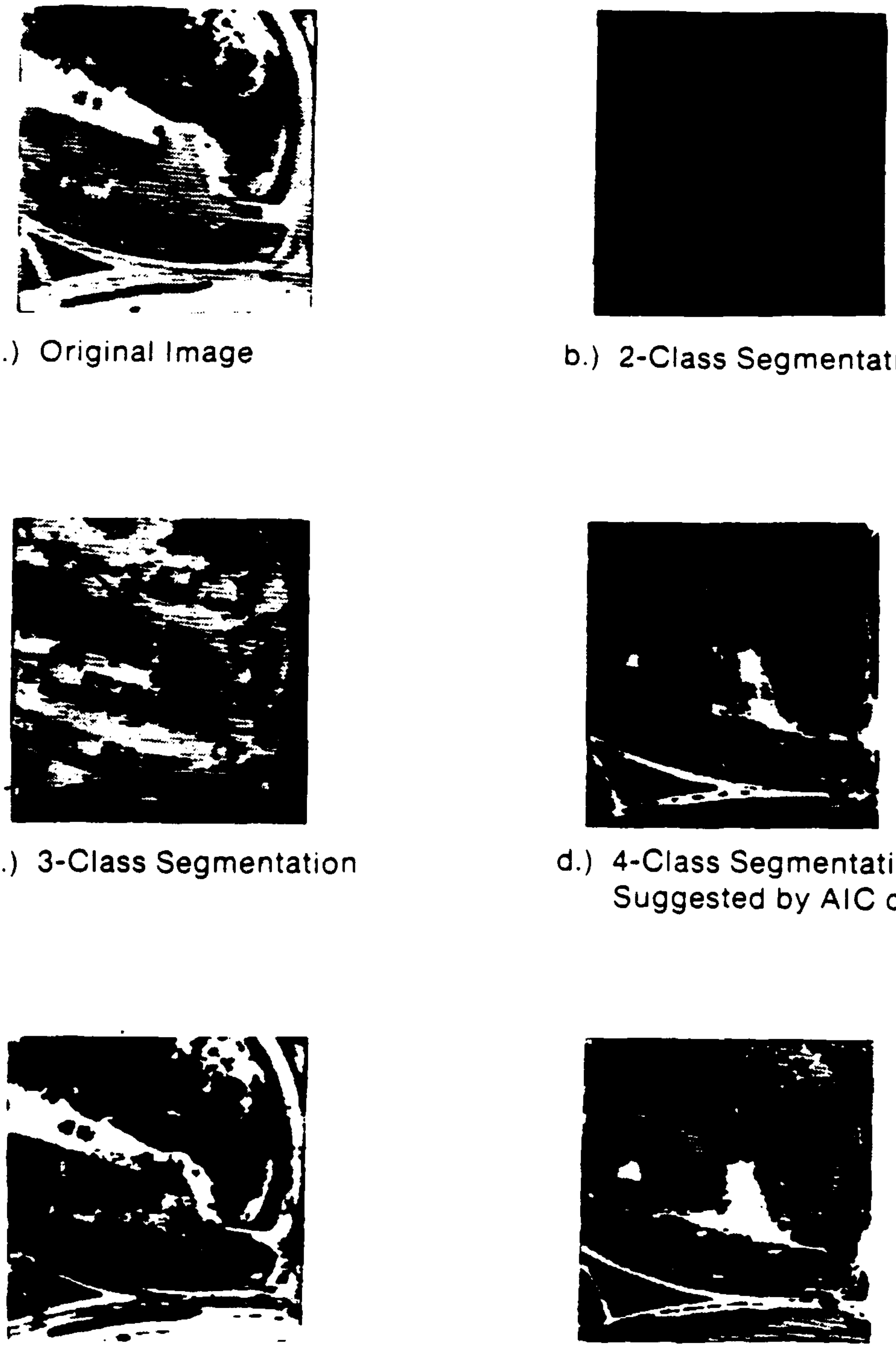


a) The 2nd and 3rd Components of the Sample AR Model Parameter Vectors

b) The Sample MRF Model Parameter Vectors

Figure 7.3.10

Sample Model Parameter Vectors for the Texture Images



a.) Original Image

b.) 2-Class Segmentation

c.) 3-Class Segmentation

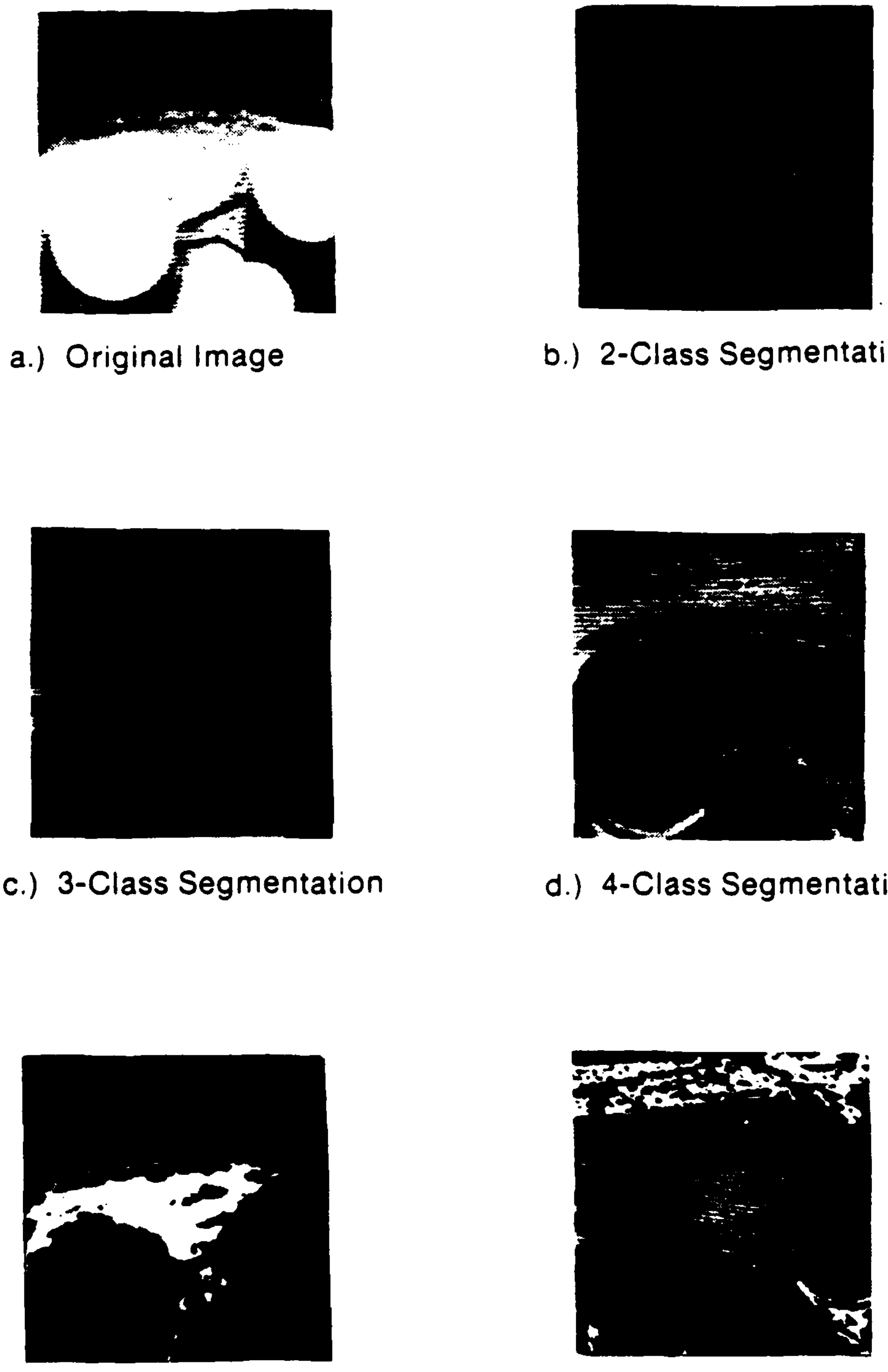
d.) 4-Class Segmentation
Suggested by AIC criterion

e.) 5-Class Segmentation

f.) 6-Class Segmentation

Figure 7.3.11

Segmented Road Scene According to the AIC criterion.



a.) Original Image

b.) 2-Class Segmentation

c.) 3-Class Segmentation

d.) 4-Class Segmentation

e.) 5-Class Segmentation
Suggested by AIC

f.) 6-Class Segmentation

Figure 7.3.12

Segmented Oil Tank Scene According to the AIC criterion.

7.4 The Photointerpretation Workstation:

7.4.1 Introduction:

The conceptual approach to automated photointerpretation developed under this effort has been described in some detail in previous sections. As mentioned in Section 7.1, we have implemented this approach in the TI/Explorer. This implementation makes use of an object-oriented based menu system and provides a fast, interactive means of performing computer vision research based on rapid clique function prototyping. The TI/Explorer implementation has been called the Photointerpretation Workstation. The remainder of this section is devoted to a description of the Photointerpretation Workstation.

7.4.2 Background

The photointerpretation workstation is a TI/Explorer based AI tool for performing computer vision research on continuous-tone images.

Our motivation for using the Explorer as a computational platform is based, in part, on the severe limitation in the data structures available in a FORTRAN programming environment. We have found the traditional data processing environment to be too restrictive.

Our objective is to move to an environment with a rich set of data structures. With the Explorer as our target system, we hope to reap the benefits of programming using an object-oriented paradigm.

7.4.3 Overview of PI Workstation

The Photo Interpretation Workstation (PIWS) is based on the TI Explorer. The PIWS is an interactive, fast, object-oriented, dedicated, hybrid test bed for photointerpretation. The hybrid nature of the PIWS comes from the TMS32030 DSP chip which is capable of 20 MFLOP of throughput on vectorized operations. In particular, we feel that this will allow for fast image segmentations. Because of the proliferation of Explorers within the NAIC, the PIWS should be readily portable.

7.4.4 User Interface

First, let's take a look at the programming environment. Using the flavor system, we are able to take advantage of the multiple-inheritance A.K.O. (A Kind Of) hierarchy.

This enables us to program in a much more flexible fashion. For example, suppose that we wanted to define a flavor, we might use a form similar to that which is presented in Fig. 7.4.1.

Here we can see that the user has defined a region flavor and made an instance of this flavor, called r1. This object, r1, may be sent a message, such as :area, and its area information is then returned. If we define a method, called :compactness, we may send the message, :compactness to the r1 instance, just as if it were an instance variable. The message is treated differently, however, since every instance of the region flavor must calculate its own compactness from given instance variables.

This is a very general approach. Let's look at how this is applied to the clique function. Recall that a typical clique function looks something like that illustrated in Fig. 7.4.2. With corner points A,B,C, and D, the clique function flavor might look like Fig. 7.4.3.

Here we see that the corner points become instance variables of the clique flavor. The user has created an instance of the clique flavor and set the atom c1 equal to it. It is possible to then define a method for the clique function which uses region-based calculations to design its own corner points. In Fig. 7.4.4, we see an example of the use of the clique flavor.

Here we see the true flexibility of the flavor system. We have defined a label flavor which contains a clique function for each feature. We broadcast to a list of regions the message feature[1] and then calculate the average of each-feature in the feature list. This is then returned as a list of feature averages whenever the label flavor is sent the :get-features message. This is of great assistance when designing clique functions.

7.4.5 Status

The PIWS does not support image processing type hardware (yet). There is no color display, no continuous-tone images, and no image digitization capability. Still we are able to display dithered images. For example, the image in Fig. 7.4.5 is displayed on a 700 by 700 pixel segment of the screen. The effective resolution is about 128 by 128 and up to 9 levels of grey are represented. Using this technique we can also represent a segmented

[1] Here broadcast is a function which sends a message to each item in the list.

image as in Fig. 7.4.6.

In Fig. 7.4.7, we see a print made from the screen of the PIWS. This PIWS frame is made up of five panes:

1. the "regions" pane,
2. the "labels" pane,
3. a "typeout window" pane,
4. the "PIWS Command Menu" pane and
5. a "status of photo" pane.

In the "regions" pane we can see the name of the region[2] followed by an assigned interpretation label. This was assigned as a result of the annealing process. The first-order energy level for the assignment follows with an overall energy for the label assignments at the bottom of the region pane. The features for each area are also present.

In the labels pane we see the clique functions with the weight for each of the assignments. Both panes are scroll windows and will scroll if the mouse is bumped up against the left hand side of the pane.

Each of the regions in the region pane is a mouse sensitive item. If the mouse comes near any of the items they are highlighted (this indicates that they are mouse sensitive). If the mouse is cliqued over region v2, the display results as in Fig. 7.4.8.

Here we can see that the vegetation label is highlighted. This indicates the *present* computer interpretation. All of the items in this pop-up menu are mouse sensitive and may be changed by the user. A similar technique is applied for the clique function pane, as indicated in Fig. 7.4.9.

7.4.6 Future Work

Currently, we input a symbolic description of the image. No segmentation is performed on the Explorer. We would like to perform all the photopreprocessing on the Explorer. This includes segmentation, histogram equalization, hand segmentation, hand interpretation, and image display. We feel that use of the TMS 32020 DSP will speed up the segmentation of the images.

[2] Note these names were assigned to a training image. They are the result of human interpretation.

The Programming Environment

- The Flavor System

  Using the flavor system:

```
(defflavor region
 (area  100)
  (.
   .
   .)
   ())

 (setq r1 (make-instance 'region))

 (send r1 :area)

 (defmethod (region :compactness) ()
  (....))
```

- This provides us with a flexible, object oriented data structure.

Figure 7.4.1

A Flavor Example

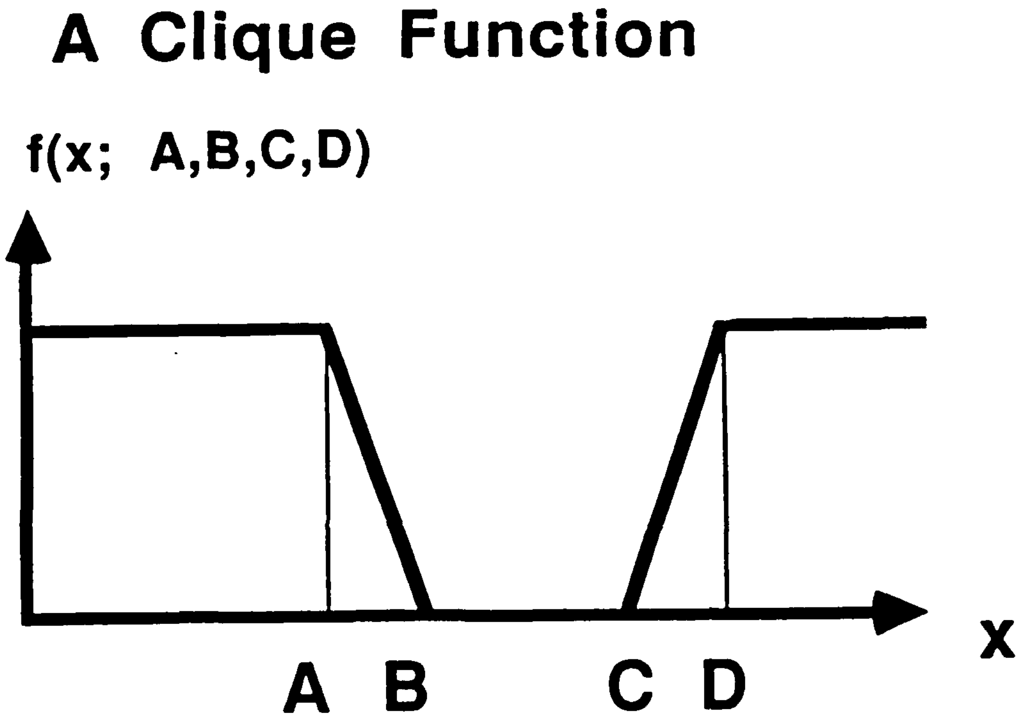
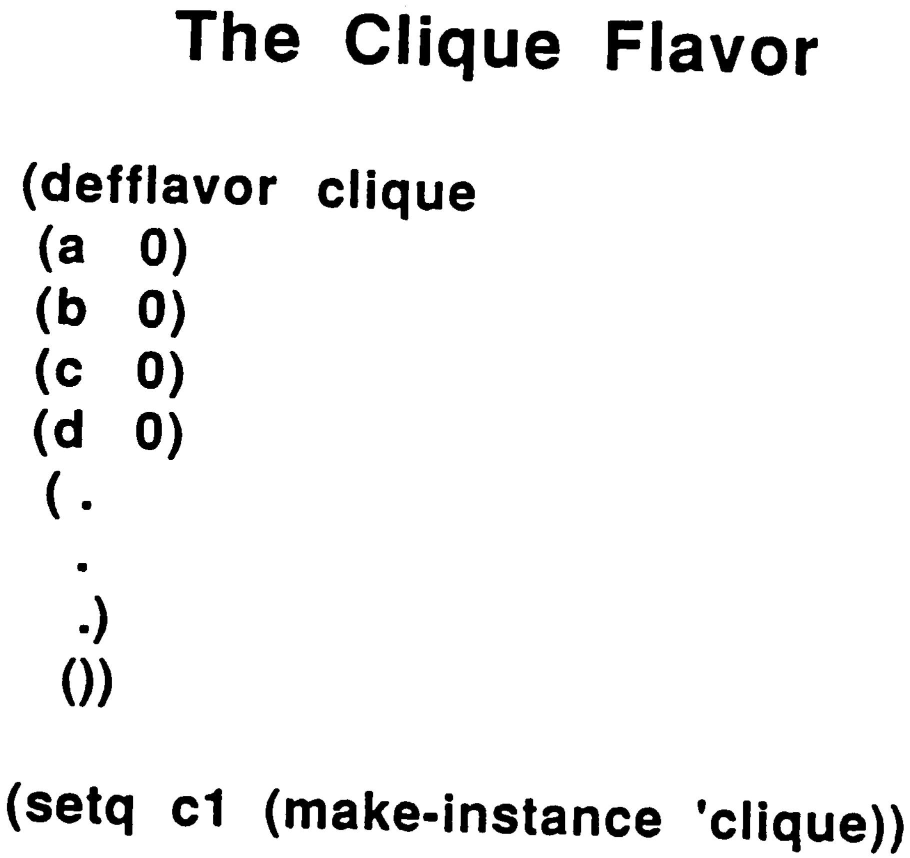


Figure 7.4.2

A Clique Function



The Clique Flavor

```
(defflavor clique
 (a  0)
 (b  0)
 (c  0)
 (d  0)
 ( .
   .
   .)
  ())

(setq c1 (make-instance 'clique))
```

Figure 7.3.3

The Clique Flavor

Using the Clique flavor

```
(setq grass (make-instance 'label))
 /* a human interpretation */
(send grass :add-regions r1 r2 r3...)
(send grass :add-clique c1)
(send grass :design-clique-function)

 (defmethod (label :design-clique-function)
             ( )
 (send self :get-features)
 (......))

 (defmethod (label :get-features) ()
  (loop for each-feature in
        feature-list do
       (send each-feature :set-average
        (average
          (broadcast regions :feature))
```

Figure 7.4.4

Using the Clique Flavor

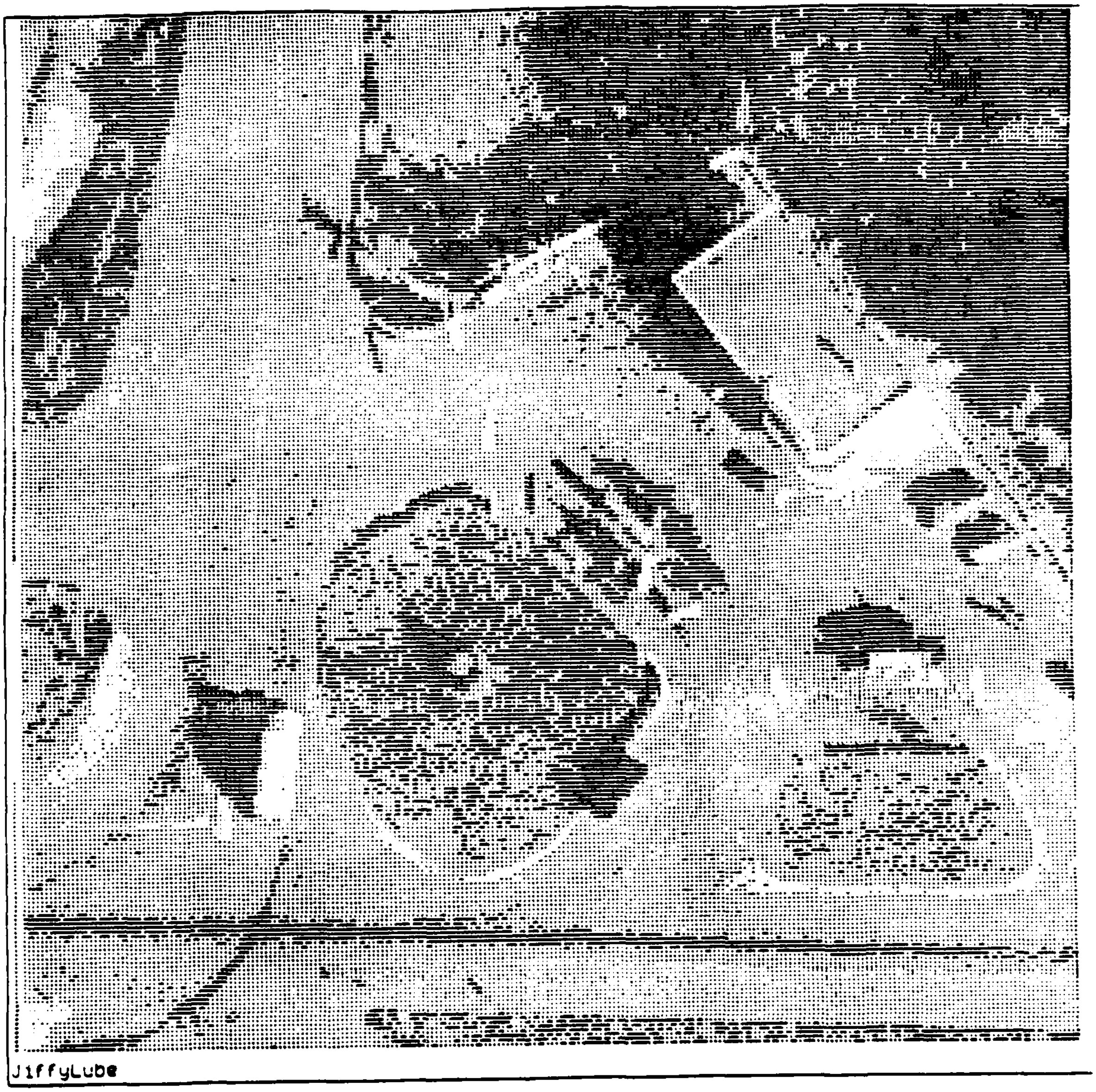


Figure 7.4.5

A Dithered Grey-Tone Image

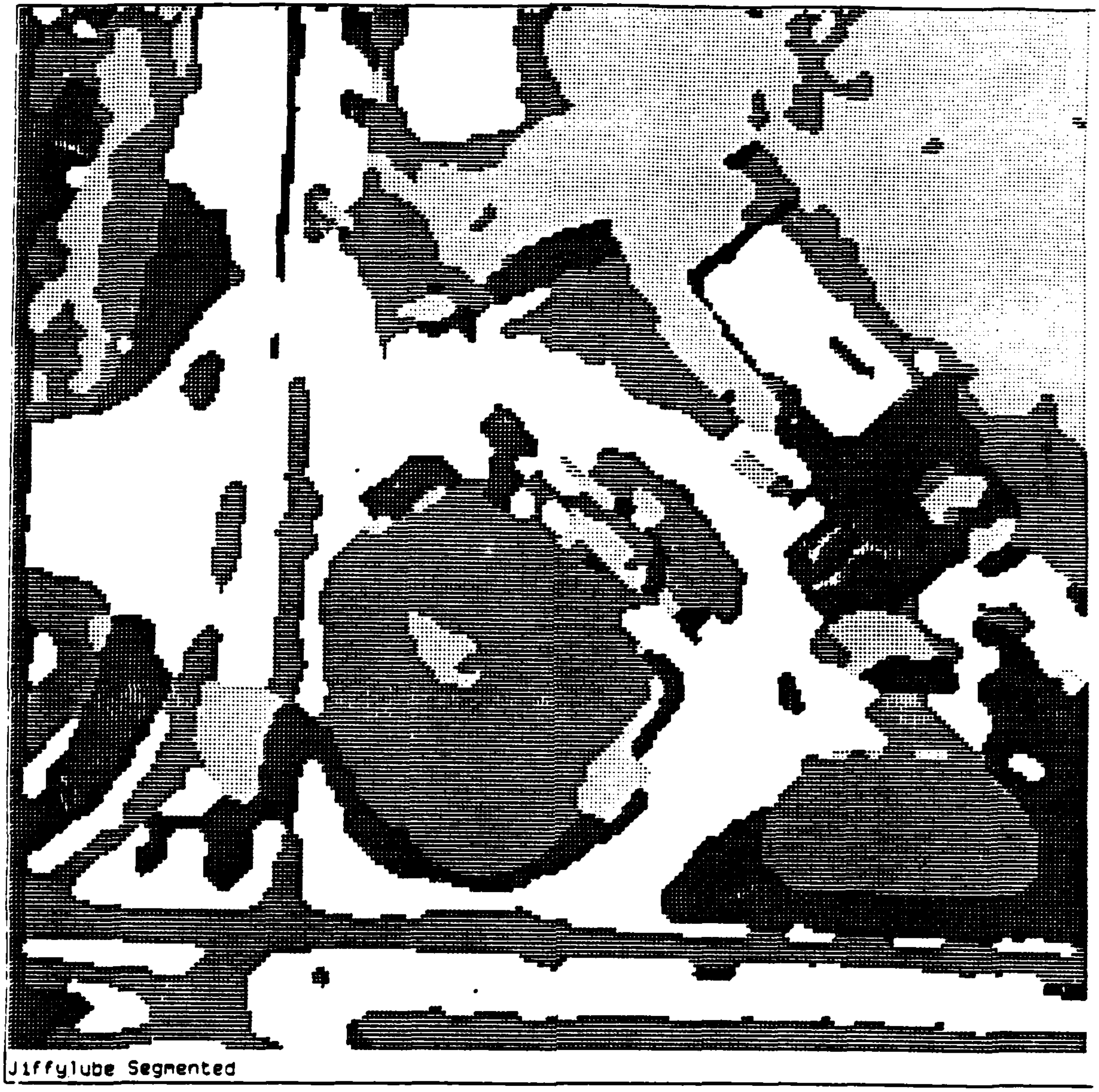
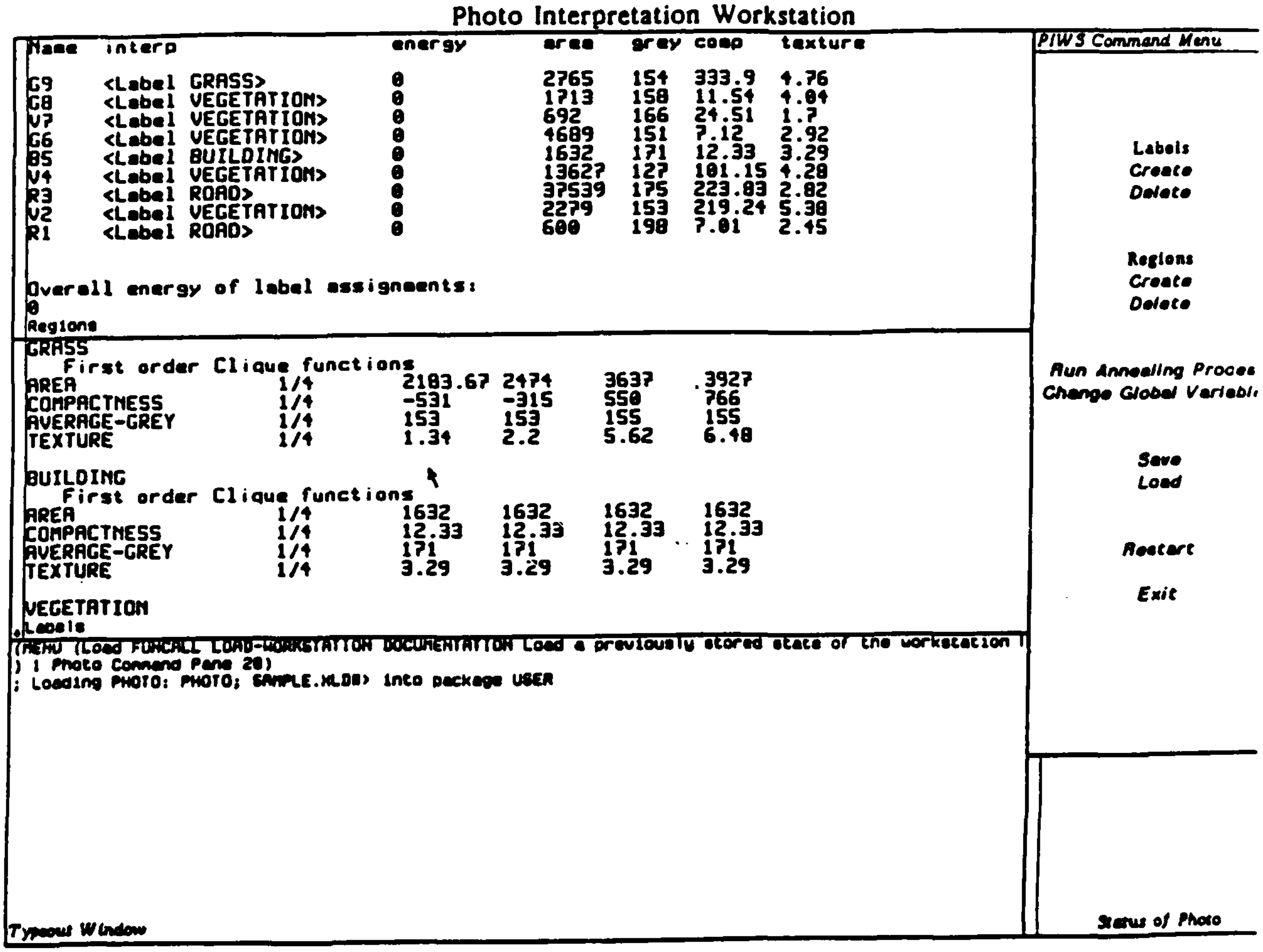


Figure 7.4.6

A Segmented Image



```
Name   interp                  energy        area   grey comp    texture

G9     <Label GRASS>           0             2765   154  333.9   4.76
G8     <Label VEGETATION>      0             1713   158  11.54   4.04
V7     <Label VEGETATION>      0             692    166  24.51   1.7
G6     <Label VEGETATION>      0             4689   151  7.12    2.92
B5     <Label BUILDING>        0             1632   171  12.33   3.29
V4     <Label VEGETATION>      0             13627  127  101.15  4.28
R3     <Label ROAD>            0             37539  175  223.83  2.82
V2     <Label VEGETATION>      0             2279   153  219.24  5.38
R1     <Label ROAD>            0             600    198  7.01    2.45

Overall energy of label assignments:
0
Regions
```

```
GRASS
   First order Clique functions
AREA                 1/4       2183.67 2474      3637      .3927
COMPACTNESS          1/4       -531    -315      550       766
AVERAGE-GREY         1/4       153     153       155       155
TEXTURE              1/4       1.34    2.2       5.62      6.48

BUILDING
   First order Clique functions
AREA                 1/4       1632    1632      1632      1632
COMPACTNESS          1/4       12.33   12.33     12.33     12.33
AVERAGE-GREY         1/4       171     171       171       171
TEXTURE              1/4       3.29    3.29      3.29      3.29

VEGETATION
Labels
```

```
(MENU (Load FUNCALL LOAD-WORKSTATION DOCUMENTATION Load a previously stored state of the workstation I
) 1 Photo Command Pane 28)
; Loading PHOTO: PHOTO; SAMPLE.XLD0> into package USER
```

Figure 7.4.7

The System Window

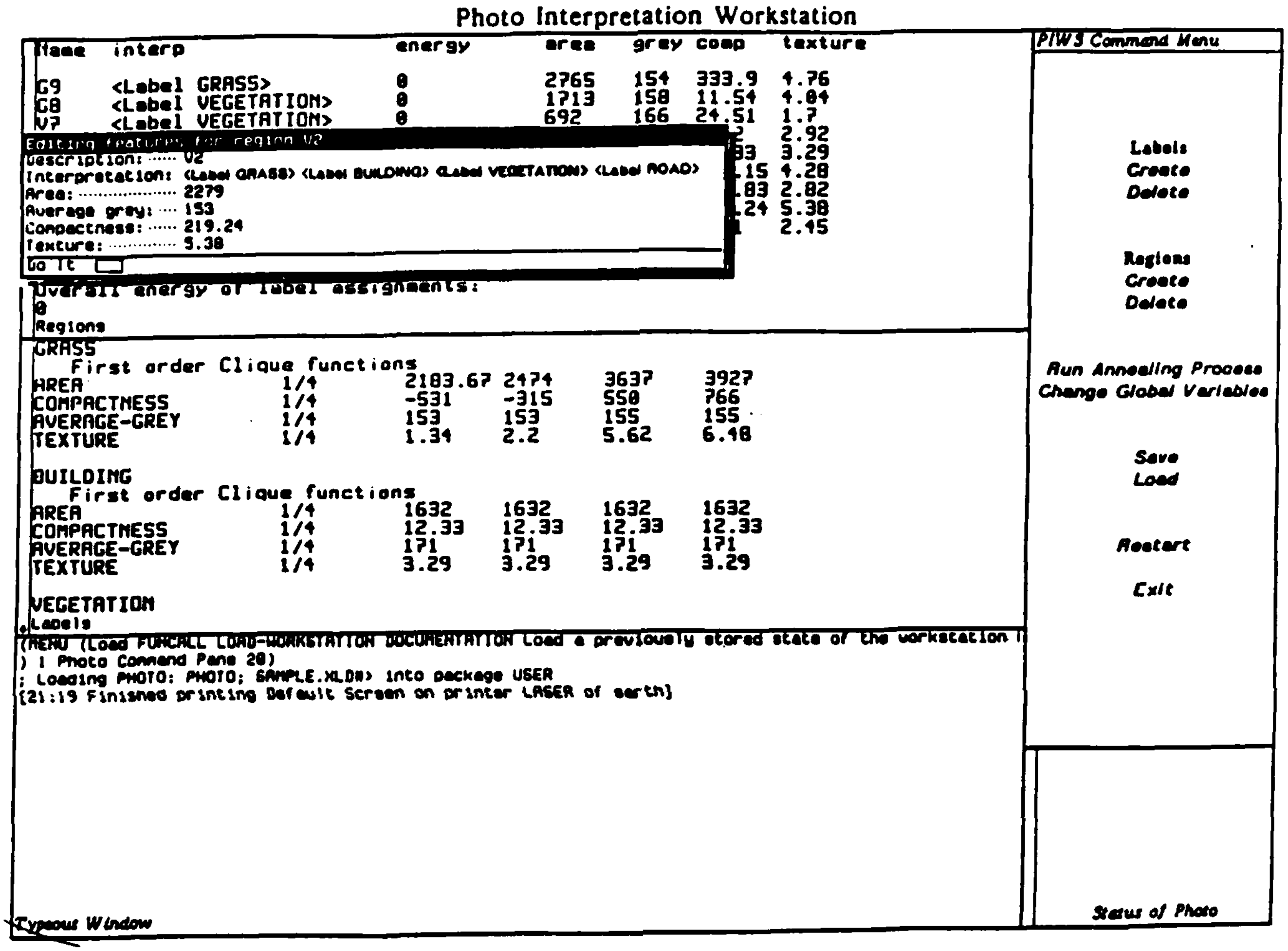


Photo Interpretation Workstation

```
Name   interp                        energy      area    grey  comp    texture

G9     <Label GRASS>                 0           2765    154   333.9   4.76
G8     <Label VEGETATION>            0           1713    158   11.54   4.04
V7     <Label VEGETATION>            0           692     166   24.51   1.7
                                                               2       2.92
                                                               33      3.29
                                                               .15     4.28
                                                               .83     2.82
                                                               .24     5.38
                                                               1       2.45
```

```
Editing features for region V2
Description: ...... V2
Interpretation: <Label GRASS> <Label BUILDING> <Label VEGETATION> <Label ROAD>
Area: ................. 2279
Average grey: .... 153
Compactness: ...... 219.24
Texture: ............ 5.38
Do It □
```

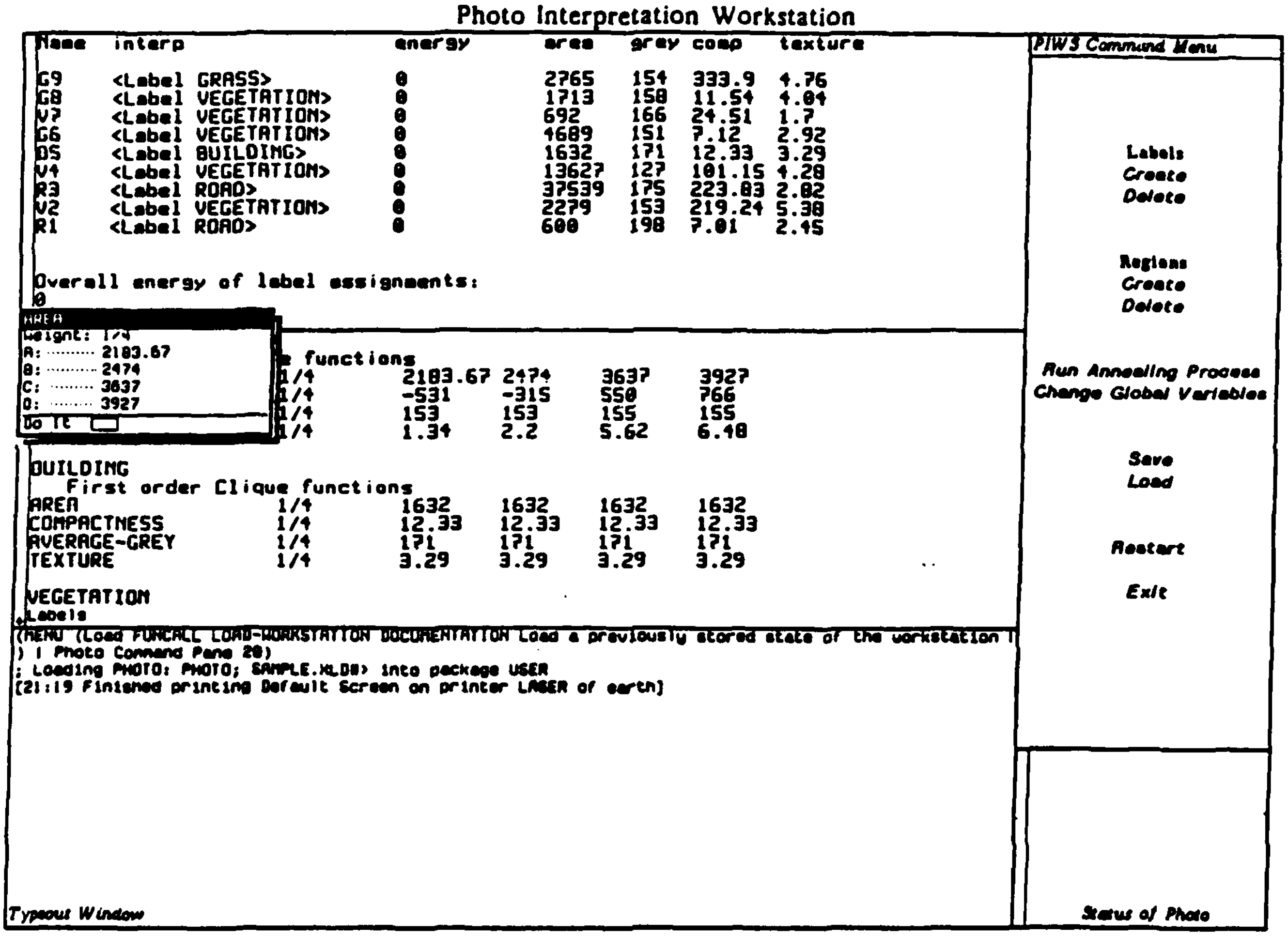
```
Overall energy of label assignments:
0
Regions
```

```
GRASS
   First order Clique functions
AREA                 1/4       2183.67 2474     3637     3927
COMPACTNESS          1/4       -531    -315     550      766
AVERAGE-GREY         1/4       153     153      155      155
TEXTURE              1/4       1.34    2.2      5.62     6.48

BUILDING
   First order Clique functions
AREA                 1/4       1632    1632     1632     1632
COMPACTNESS          1/4       12.33   12.33    12.33    12.33
AVERAGE-GREY         1/4       171     171      171      171
TEXTURE              1/4       3.29    3.29     3.29     3.29

VEGETATION
Labels
```

```
(MENU (Load FUNCALL LOAD-WORKSTATION DOCUMENTATION Load a previously stored state of the workstation (
) 1 Photo Command Pane 20)
; Loading PHOTO: PHOTO; SAMPLE.XLD#> into package USER
[21:19 Finished printing Default Screen on printer LASER of earth]
```

Figure 7.4.8

Editing Features for a Region



```
Name   interp                      energy       area    grey comp    texture

G9     <Label GRASS>               0            2765    154  333.9   4.76
G8     <Label VEGETATION>          0            1713    158  11.54   4.04
V7     <Label VEGETATION>          0            692     166  24.51   1.7
G6     <Label VEGETATION>          0            4689    151  7.12    2.92
D5     <Label BUILDING>            0            1632    171  12.33   3.29
V4     <Label VEGETATION>          0            13627   127  101.15  4.28
R3     <Label ROAD>                0            37539   175  223.83  2.82
V2     <Label VEGETATION>          0            2279    153  219.24  5.38
R1     <Label ROAD>                0            600     198  7.01    2.45

Overall energy of label assignments:
0
```

```
AREA
Weight: 1/4
A: ......... 2183.67
B: ......... 2474
C: ......... 3637
D: ......... 3927
```

```
e functions
1/4          2183.67 2474      3637      3927
1/4          -531      -315    550       766
1/4          153       153     155       155
1/4          1.34      2.2     5.62      6.48

BUILDING
   First order Clique functions
AREA                1/4        1632      1632      1632      1632
COMPACTNESS         1/4        12.33     12.33     12.33     12.33
AVERAGE-GREY        1/4        171       171       171       171
TEXTURE             1/4        3.29      3.29      3.29      3.29

VEGETATION
Labels
```

```
(MENU (Load FUNCALL LOAD-WORKSTATION DOCUMENTATION Load a previously stored state of the workstation !
) 1 Photo Command Pane 20)
; Loading PHOTO: PHOTO; SAMPLE.XLD#> into package USER
[21:19 Finished printing Default Screen on printer LASER of earth]
```

Figure 7.4.9

Editing a Clique Function